A PARAMETER OPTIMIZATION APPROACH TO AIRCRAFT GUST ALLEVIATION

Russell Lee Clement

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A PARAMETER OPTIMIZATION APPROACH TO
AIRCRAFT GUST ALLEVIATION

by

Russell Lee Clement

Thesis Advisor: Ronald A. Hess

March 1972

A Parameter Optimization Approach to Aircraft Gust Alleviation

by

Russell Lee Clement
Lieutenant Commander, United States Navy
B.S., Naval Postgraduate School, 1965

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1972

ABSTRACT

This research indicates that a knowledge of angle of attack (α) and/or gust perturbation angle of attack (α_g) is of prime importance for a longitudinal autopilot designed to alleviate the effect of vertical gusts. However, accurate measurement of α and α_g is extremely difficult. This research indicates that the readily measurable normal acceleration factor (n) and pitch rate ($\hat{q}$) can be used to provide α and α_g information indirectly.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SYMBOLS

a_{c_z} = Linear acceleration of aircraft center of mass in z direction, ft/sec.

$\bar{c}$ = Length of mean aerodynamic chord, ft.

c_{m_η} = $\left.\frac{\partial c_m}{\partial \eta}\right|_o$

c_{L_α} = $\left.\frac{\partial c_L}{\partial \alpha}\right|_o$

g = Local acceleration of gravity, ft/sec.

h = Altitude, (ft).

$\overline{e_o^2}$ = Mean-square value of output of system.

$K1 \rightarrow K4$ = Control system gains.

L_w = Gust characteristic length, ft.

m = Aircraft mass, Slugs.

n, $\overline{n^2}$ = a_{c_z}/g, normal acceleration factor, ft/sec^2/g.

$\Delta\overline{n^2}$ = $\overline{n^2}_{unallev.} - \overline{n^2}$, (ft/sec^2/g)^2.

$\overline{n^2}_{unallev.}$ = Value of $\overline{n^2}$ with no autopilot utilized, (ft/sec^2/g)^2.

$\hat{q}$ = Pitch rate, rad/airsec.

$\hat{q}_g$ = Pitch rate due to vertical gust, rad/airsec.

U_o = Aircraft forward velocity, ft/sec.

u = Aircraft forward velocity perturbation, ft/sec.

xyz = Stability axis system.

w = Velocity perturbation along z axis, ft/sec.

w_g = Gust vertical velocity, ft/sec.

α_g = w_g/U_o, angle of attack perturbation due to gust, measured at aircraft center of mass, rad.

η = Elevator deflection, measured from trim, rad.

$\overline{\eta^2}$ = Mean-square value of elevator deflection $(\text{Rad})^2$.

δ_f = DLC (direct lift control) deflection.

δ_t = Throttle deflection.

Ω = Spatial frequency, rad/ft.

$\hat{\omega}$ = $\Omega U_o t^* = \Omega U_o \frac{\overline{C}}{2U_o}$, Frequency, Rad/AirSec.

σ_w = Root mean-square value of vertical gust velocity, ft/sec.

θ = Pitch angle perturbation, Rad.

ρ = Atmospheric density, Slugs/Ft^3.

ACKNOWLEDGEMENTS

My most sincere appreciation to Dr. Ronald A. Hess for his patience and professional guidance throughout this research.

I. INTRODUCTION

A. LONGITUDINAL AUTOPILOTS

1. Basic Ideas in Longitudinal Control

Since the beginning of powered flight, methods have been sought to reduce pilot effort in controlling aircraft and to automate various aspects of aircraft control. The autopilots and stability augmentation systems of today are realizations of such research.

Autopilots and stability augmentation systems are usually classified as longitudinal or lateral. As the names imply, longitudinal (lateral) autopilots are concerned with the longitudinal (lateral) flight regulation. This dual classification is justified in flight conditions where the existence of pure longitudinal/lateral motion is ensured.

Blakelock [Ref. 1] offers a very readable introduction to autopilot design. He discusses three basic types of longitudinal controllers, each designed for different longitudinal operations: a displacement autopilot where aircraft pitch angle is fed back and compared with a reference pitch angle; a pitch orientational autopilot where aircraft pitch rate is fed back and compared with some desired pitch rate; and an acceleration control system where the acceleration of the center of gravity is fed back and compared with a reference acceleration. A good treatment of the feedbacks that may be used in longitudinal control systems can be found in Ref. 2, Sec. II. Table I (obtained from Ref. 2) indicates the different feedbacks which can be utilized to control longitudinal motion.

The research summarized here is concerned with autopilots designed to alleviate the effect of vertical gusts on a large jet transport in steady, wings-level flight. Two flight conditions are considered, high altitude cruise and landing approach. By reducing the aircraft's gust response it is possible to provide for a smoother ride, aircraft structural fatigue problems can be alleviated, and the ability of the aircraft to function as a gun platform can be imporved.

2. Longitudinal Control to Alleviate Gusts

Appendix B offers an introduction to the statistical design concepts utilized in this study. As Table I indicates many feedback parameters such as pitch, pitch rate, center of gravity acceleration, etc., can be utilized as feedbacks in a controller designed to reduce gust response. Hess [Ref. 3] discussed a two-parameter gust alleviation system shown in Fig. 1. He analyzed this system using a performance index given by:

$$I = \frac{1}{2\pi} \int_{\infty}^{\infty} \Phi_{\alpha g}(\hat{\omega}) \left| \frac{n}{\alpha_g}(s) \right|^2_{s=j\hat{\omega}} d\hat{\omega} + \int_{\infty}^{\infty} \Phi_{\alpha g}(\hat{\omega}) \left| \frac{n}{\alpha_g}(s) \right|^2_{s=j\hat{\omega}} d\hat{\omega}$$

where $\hat{\omega}$ is the non-dimensionalized frequency and the transfer functions $\frac{n}{\alpha_g}(s)$ and $\frac{n}{\alpha_g}(s)$ are obtained from the aircraft's equation of motion as discussed in Appendix C. $\Phi_{\alpha_g}(\hat{\omega})$ is the non-dimensionalized power spectral density of the gust field and was obtained from Equation 1 found in Appendix B. Hess's study utilized a large, rigid jet transport. The characteristics for the two flight conditions considered may be found in Appendix A. His results are shown in Table II.

Hess also proposed an optimal control system, Fig. 2, which showed considerable improvement over the two-parameter feedback system above. For analysis of this system Hess used a performance index given by:

$$I = \lim_{T\to\infty} \frac{1}{T} \int_0^T [qn^2(t) + ru^2(t)]dt$$

The aircraft was the same as used in analysis of the two-parameter system. Table III shows results of his study for a q/r equal to 10, a value was selected by a root-square locus technique. Note the improvement in $\frac{\overline{\Delta n^2}}{\overline{n^2}_{unalleviated}}$ which Hess's optimal system maintains over the two-parameter system of Table II.

B. BASIS OF THIS RESEARCH

A major obstacle to the implementation of Hess's optimal gust alleviation system is the difficulty in measuring α_g. Attempts to design instruments that will give satisfactory on-line measurements of α_g have been unsuccessful. This report investigated the effect of eliminating α_g information on the performance of the alleviation system. Figure 3 shows Hess's optimal gust alleviation system with gust information removed. This is the system utilized in this study.

II. SOLUTION PROCEDURES

A. METHODS OF SOLUTION

Optimal feedback design can be classified as free configuration or fixed configuration. The latter is often referred to as parameter optimization. In the free configuration procedure the designer is not restricted in any way. The final design depends only upon the input and the form of the performance criteria. With the parameter optimization techniques the configuration is chosen using feedbacks which are readily obtainable and then the feedback parameters are optimized to minimize the performance criteria.

As noted in Ref. 2, for autopilot design the parameter optimization technique is probably the superior method considering current state of the art. This technique yields an autopilot which fits the "real world" situation. This research utilized the parameter optimization procedure since the form of the longitudinal controller was prespecified (Fig. 3).

B. PERFORMANCE INDEX

The performance index chosen for the optimization procedure was given by:

$$I = \frac{1}{2\pi}\left[\int_{\infty}^{\infty}\Phi_n(\hat{\omega})\,d\hat{\omega} + \int_{\infty}^{\infty}\Phi_\eta(\hat{\omega})\,d\hat{\omega}\right] = \overline{n^2} + \overline{\eta^2} \qquad (1)$$

where $\hat{\omega}$ denotes the non-dimensionalized frequency of the random gusts. This performance index is identical to that used by Hess in his evaluation of the optimal controller with $q/r = 10$. It is a sum of the mean square

normal acceleration factor and the mean square elevator deflection. The latter was included to prevent large elevator excursions. Equation (1) can be manipulated as indicated in Appendix B to a more usable form:

$$I = \frac{1}{2\pi}\int_{\infty}^{\infty} \Phi_{\alpha g}(\hat{\omega}) \left|\frac{n}{\alpha_g}(j\hat{\omega})\right|^2 d\hat{\omega} + \frac{1}{2\pi}\int_{\infty}^{\infty} \Phi_{\alpha g}(\hat{\omega}) \left|\frac{n}{\alpha_g}(j\hat{\omega})\right|^2 d\hat{\omega} \qquad (2)$$

$\Phi_{\alpha g}$ is obtained from Equation (1) in Appendix B by substituting $\hat{\omega} = \Omega u_o t^*$. This non-dimensionalized form is:

$$\Phi_{\alpha g}(\hat{\omega}) = \frac{\sigma_w^2 \bar{c}}{L_w U_o^2} \quad \frac{1}{\hat{\omega}^2 + (\bar{c}^2/4L_w^2)} \qquad (3)$$

The transfer functions $\frac{n}{\alpha_g}(j\hat{\omega})$ and $\frac{n}{\alpha_g}(j\hat{\omega})$ are obtained from the aircraft's longitudinal equations of motion, the equation for normal acceleration and the mathematical equation of the autopilot. Appendix C shows how this was done.

For the purposes of determining the aerodynamic gust effects, the aircraft was idealized as a segment of the x-axis. This idealization has been found to give good results as long as the shortest vertical-gust wave length is greater or equal to eight times the aircraft length [Ref. 4, pg. 323].

C. INTEGRATION OF THE PERFORMANCE INDEX

Substitution of Equation (3) and the transfer functions into Equation (2) produced an equation whose form is frequently found in system analysis. Tables for solution of these equations are readily available, e.g. Ref. 5.

Appendix D shows how a computer program was written to solve for the performance index, which in this case took on the functional form of $I_4(K1,K2,K3,L_W)$.

D. OPTIMIZATION

Integration of the performance index for a given characteristic gust length (L_W) produced a function which contained the variables K1, K2, and K3, i.e. the feedback gains. The function obtained could be thought of as a performance index surface with dimensions K1, K2 and K3. The optimum solution would be found where the surface produced a minimum value. A computer program was written to solve for this minimum by a method of steepest descent. Appendix D indicates how this was accomplished.

III. RESULTS AND DISCUSSION

A. PERFORMANCE COMPARISONS

The results of the computer computation are presented in Table IV. The performance surface at most points was very shallow making it difficult to rapidly find a minimum value on the surface. It was found that the performance index was a weak function of K2, but a strong function of K1, K3 and L_w. K1, K2 and K3 were found to be very dependent on the flight condition, i.e. cruise or landing approach.

Comparing Table IV with Tables II and III shows that the three-parameter system has the largest performance index. Figures 5 and 6 indicate that the $\Delta\overline{n^2}/\overline{n^2}_{unallev.}$ performance initially compares favorably with the optimal and the two-parameter system, but the performance rapidly deteriorates as L_w increases. This indicates that a knowledge of α_g becomes increasingly important as L_w gets larger.

Note that in the two-parameter system of Fig. 1 the feedbacks n and $\hat{q}$ give a knowledge of α and α_g. This can be seen from the equation for the normal acceleration factor which can be written as:

$$D(\alpha + \alpha_g) = \hat{q} - \frac{g\bar{c}}{2U_o^2} n$$

B. TRANSIENT PERFORMANCE

Figures 6 and 7 present the locus of roots of the characteristic equation obtained when the feedback gains K1 and K3 are set at values obtained for a gust characteristic length of 1000 feet while K2 was allowed to vary. This could be done because the performance index was

a weak function of K2. A value of the variable K2 could be selected which would give the three-parameter system better transient performance than the two-parameter system, but this transient performance was still inferior to the optimal control system. A new root locus diagram would have to be drawn for each gust length and flight condition. In an effort to circumvent the drawing of multiple root locus diagrams, the feedback gains K1 and K3 were set at values obtained for L_w equal to 3000 feet. It is found, however, that by doing this performance was unacceptably degraded, particularly in the lower ranges of L_w.

C. STABILITY

It was found that the three-parameter system was stable throughout the range of feedback gains considered here.

D. POSSIBLE ALTERNATE SOLUTION

Comparing the three-parameter longitudinal controller of this research with the optimal controller (Fig. 2) and the two-parameter system (Fig. 1) it appears that α_g is an important parameter for a gust alleviation system. Since, using present technology, it is not possible to measure α_g aboard the aircraft some alternative method of obtaining this information must be employed.

It was noted previously (Sec. III, Part A) that a knowledge of n and $\hat{q}$ would provide information on α and α_g. This would explain why the RMS performance of the two parameter feed back system tends to approach the RMS performance of the optimal controller as the characteristic gust length increases.

It is felt that by combining the optimal controller and the two-parameter system a gust alleviation system could be built that would have performance comparable to the optimal controller. Indeed, Hess [Ref. 6] has shown this to be the case. The modified gust alleviation system appears in Fig. 4. This system would have the advantage of utilizing feedback information that is readily obtainable on-board the aircraft.

IV. CONCLUSIONS

1. The longitudinal gust alleviation system evaluated by this report is a stable, usable system and could be designed to provide better transient performance than the two-parameter system of Fig. 1. However, the two-parameter system and the optimal controller of Fig. 2 both exhibit better RMS performance.

2. α_g provides important information for a gust alleviation system. Knowledge of α_g becomes increasingly important as the gust characteristic length increases, i.e. as the random gust field departs from white noise.

3. To provide for a gust alleviation system which uses readily obtainable information, and still produce satisfactory performance, the system presented in Fig. 4 is proposed.

Table I
LONGITUDINAL COMPETING SYSTEMS

FEEDBACK OR CROSSFEED	PRIMARY FUNCTIONS PERFORMED	PRACTICAL DESIGN PROBLEMS
$\hat{\underline{q}} \rightarrow \eta$	1. Increase short-period damping. 2. Reduce pitch response to gusts.	Gain adjustment with flight condition.
$\theta \rightarrow \eta$	1. Increase short-period damping and frequency. 2. Reduce pitch response to gusts. 3. Increase phugoid damping. 4. Stabilize tuck mode.	1. Gain adjustment with flight condition. 2. Increased h and a_z response to vertical gusts.
$a_{c_z} \rightarrow \eta$	1. Increase short-period damping and frequency. 2. Reduce h and a_z response to gusts.	1. Severe gain adjustment with flight condition. 2. Sensor location adequate for all flight conditions. 3. Structural mode feedback. 4. Increased θ response to vertical gusts.
$\alpha \rightarrow \eta$	1. Increase short-period damping and frequency. 2. Reduce h and a_z response to vertical gusts.	1. Gain adjustment with flight conditions. 2. Sensor instrumentation. a. Determination of operating point. b. Errors due to aerodynamic interference. c. Elaborate sensor complex required to suppress gust inputs.
$u \rightarrow \eta$	Stabilize tuck mode.	Sensor instrumentation (see above).

Table I (Continued)
LONGITUDINAL COMPETING SYSTEMS

FEEDBACK OR CROSSFEED	PRIMARY FUNCTIONS PERFORMED	PRACTICAL DESIGN PROBLEMS
$\alpha \rightarrow \delta_T$	Prevent altitude instability.	Sensor instrumentation (see above).
$u \rightarrow \delta_T$	Prevent altitude instability.	Sensor instrumentation (see above).
$a_{c_z} \rightarrow \delta_F$	Reduce h and a_z response to gusts.	1. Severe gain adjustment with flight condition. 2. Sensor location adequate for all flight conditions. 3. Structural mode feedback. 4. Probable drag penalty due to direct lift control surface.
$n \rightarrow \delta_T$	Prevent altitude instability.	Will not work if stick force/knot gets too low.

Table II
TWO-PARAMETER LONGITUDINAL AUTOPILOT

Cruising Flight

K_1 Rad/Ft/Airs^2/g	K_2 Rad/Rad/Airs	L_w Ft	$\overline{n^2}$ (Ft/Sec^2/g)^2	$\overline{\eta^2}$ (Rad)^2	$\frac{\overline{\Delta n^2}}{\overline{n^2}}$ unallev.	σ_w (Ft/Sec)
.36	675	500	.0466	.00282	.27	10
.36	475	1000	.0277	.00153	.37	10
.36	400	2000	.0151	.000818	.41	10
.36	375	3000	.0139	.000501	.24	10
.36	350	4000	.00790	.000435	.44	10
.36	350	5000	.00637	.000348	.44	10
.36	350	6000	.00534	.000290	.44	10

Landing Approach

K_1	K_2	L_w	$\overline{n^2}$	$\overline{\eta^2}$	$\frac{\overline{\Delta n^2}}{\overline{n^2}}$ unallev.	σ_w
.74	275	500	.0384	.0138	.21	10
.74	200	1000	.0209	.00690	.30	10
.74	175	2000	.0109	.00351	.41	10
.74	175	3000	.00736	.00234	.37	10
.74	175	4000	.00556	.00176	.38	10
.74	175	5000	.00447	.00141	.38	10
.74	175	6000	.00373	.00117	.39	10

Table III
OPTIMAL LONGITUDINAL CONTROLLER
Cruising Flight

q/r	L_w Ft	K_1 Rad/Rad	K_2 Rad/Rad/Airs	K_3 Rad/Rad	K_4 Rad/Rad	$\overline{n^2}$ $(Ft/Sec^2/g)^2$	$\overline{\eta^2}$ $(Rad)^2$	$\overline{\Delta n^2}/\overline{n^2}_{Unall.}$	σ_w Ft/Sec
10.0	500	44.06	1231.	-3.454	37.4	4.008 10^{-2}	2.790 10^{-3}	.37	10
10.0	1000	44.06	1231.	-3.454	37.4	2.252 10^{-2}	1.441 10^{-3}	.48	10
10.0	2000	44.06	1231.	-3.454	37.4	1.226 10^{-2}	7.303 10^{-4}	.53	10
10.0	3000	44.06	1231.	-3.454	37.4	8.589 10^{-3}	4.891 10^{-4}	.53	10
10.0	4000	44.06	1231.	-3.454	37.4	6.702 10^{-3}	3.679 10^{-4}	.52	10
10.0	5000	44.06	1231.	-3.454	37.4	5.552 10^{-3}	2.950 10^{-4}	.51	10
10.0	6000	44.06	1231.	-3.454	37.4	4.779 10^{-3}	2.463 10^{-4}	.50	10

Table III (Continued)
OPTIMAL LONGITUDINAL CONTROLLER
Landing Approach

q/r	L_w Ft	K_1 Rad/Rad	K_2 Rad/Rad/Airs	K_3 Rad/Rad	K_4 Rad/Rad	$\overline{n^2}$ $(Ft/Sec^2/g)^2$	$\overline{\eta^2}$ $(Rad)^2$	$\frac{\overline{\Delta n^2}}{\overline{n^2}}$ Unall.	σ_w Ft/Sec
10.0	500	15.02	279.2	-1.253	13.1	3.143 10^{-2}	8.482 10^{-3}	.35	10
10.0	1000	15.02	279.2	-1.253	13.1	1.712 10^{-2}	4.369 10^{-3}	.43	10
10.0	2000	15.02	279.2	-1.253	13.1	9.097 10^{-3}	2.219 10^{-3}	.46	10
10.0	3000	15.02	279.2	-1.253	13.1	6.282 10^{-3}	1.490 10^{-3}	.46	10
10.0	4000	15.02	279.2	-1.253	13.1	4.847 10^{-3}	1.123 10^{-3}	.46	10
10.0	5000	15.02	279.2	-1.253	13.1	3.997 10^{-3}	9.017 10^{-4}	.45	10
10.0	6000	15.02	279.2	-1.253	13.1	3.393 10^{-3}	7.542 10^{-4}	.44	10

Table IV
THREE PARAMETER LONGITUDINAL CONTROLLER
Cruising Flight

L_W Ft.	K1 Rad/Rad	K2 Rad/Rad / Airs	K3 Rad/Rad	$\overline{n^2}$ $(Ft/Sec^2/g)^2$	$\overline{\eta^2}$ $(Rad)^2$	$\frac{\overline{\Delta n^2}}{\overline{n^2}_{unall}}$
500	1.59	688.0	-2.57	.0461	.0001	.276
1000	1.60	688.0	-2.57	.0324	.0001	.259
2000	0.795	688.0	-3.08	.0207	*	.201
3000	0.568	688.0	-3.07	.0150	*	.176
4000	0.452	688.0	-2.99	.0120	*	.143
5000	0.393	688.0	-2.96	.0100	*	.123
6000	0.447	688.0	-2.99	.0085	*	.115

σ = 10 Ft/Sec

* - Value less than .0001

Landing Approach

L_W Ft.	K1 Rad/Rad	K2 Rad/Rad / Airs	K3 Rad/Rad	$\overline{n^2}$ $(Ft/Sec^2/g)^2$	$\overline{\eta^2}$ $(Rad)^2$	$\frac{\overline{\Delta n^2}}{\overline{n^2}_{unall}}$
500	0.651	400.0	-1.00	.0356	.0011	.262
1000	0.785	400.0	-1.13	.0219	.0008	.270
2000	0.255	400.0	-0.996	.0124	.0004	.250
3000	0.165	400.0	-0.939	.0086	.0003	.265
4000	0.165	400.0	-0.939	.0066	.0003	.236
5000	0.098	400.0	-0.974	.0054	.0002	.250
6000	0.085	400.0	-0.953	.0045	.0002	.250

σ = 10 Ft/Sec

FIGURE 1

Two Parameter Gust Alleviation System

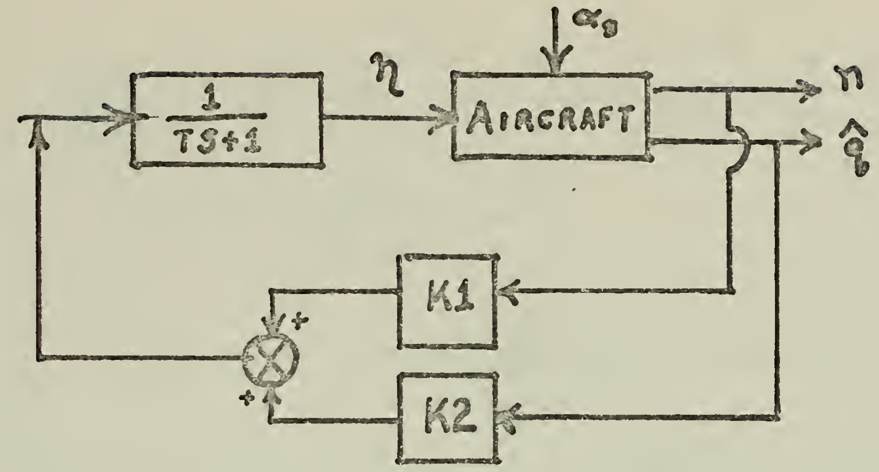


FIGURE 2

Optimum Gust Alleviation System

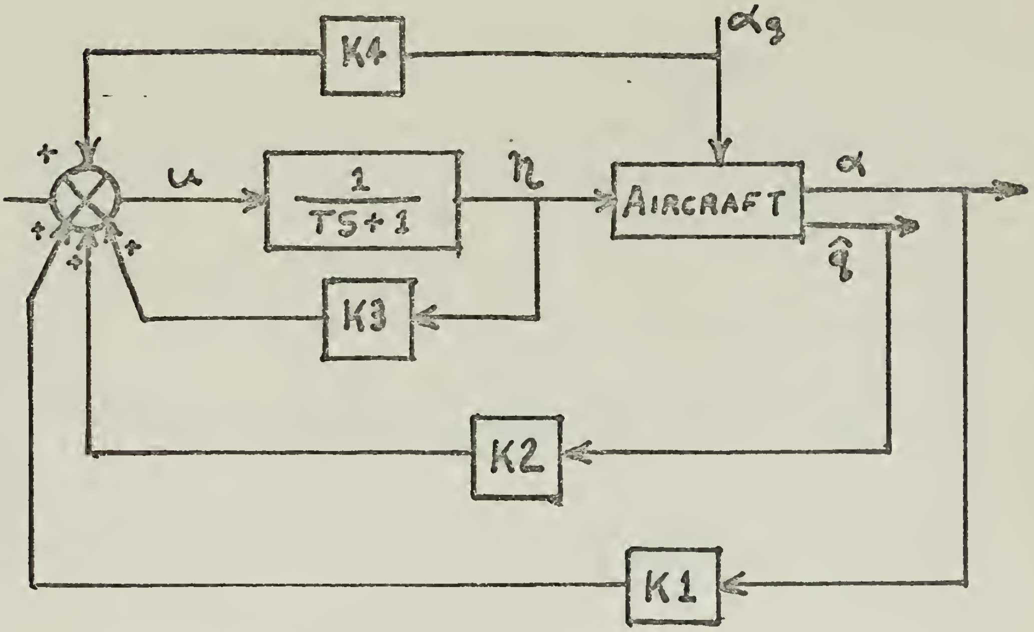


FIGURE 3

Three Parameter Gust Alleviation System

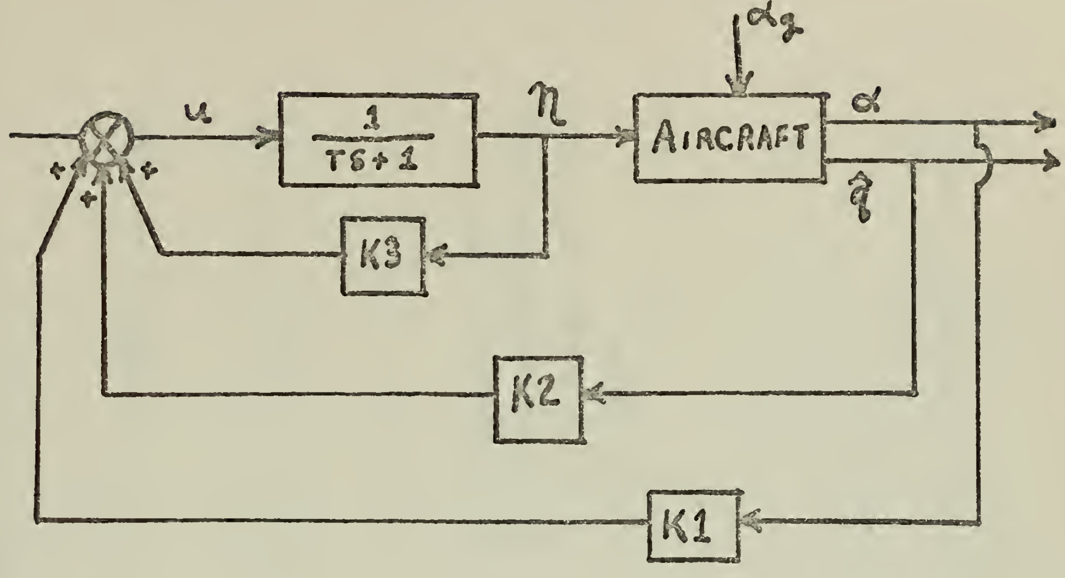


FIGURE 4

Proposed Gust Alleviation System

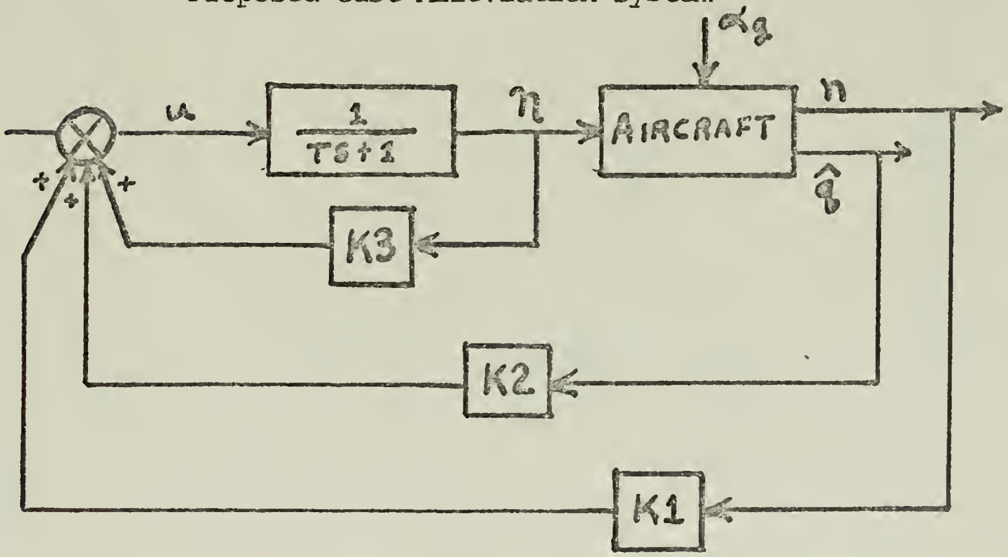


FIGURE 5. PERFORMANCE INDEX COMPARISONS

Cruising Flight

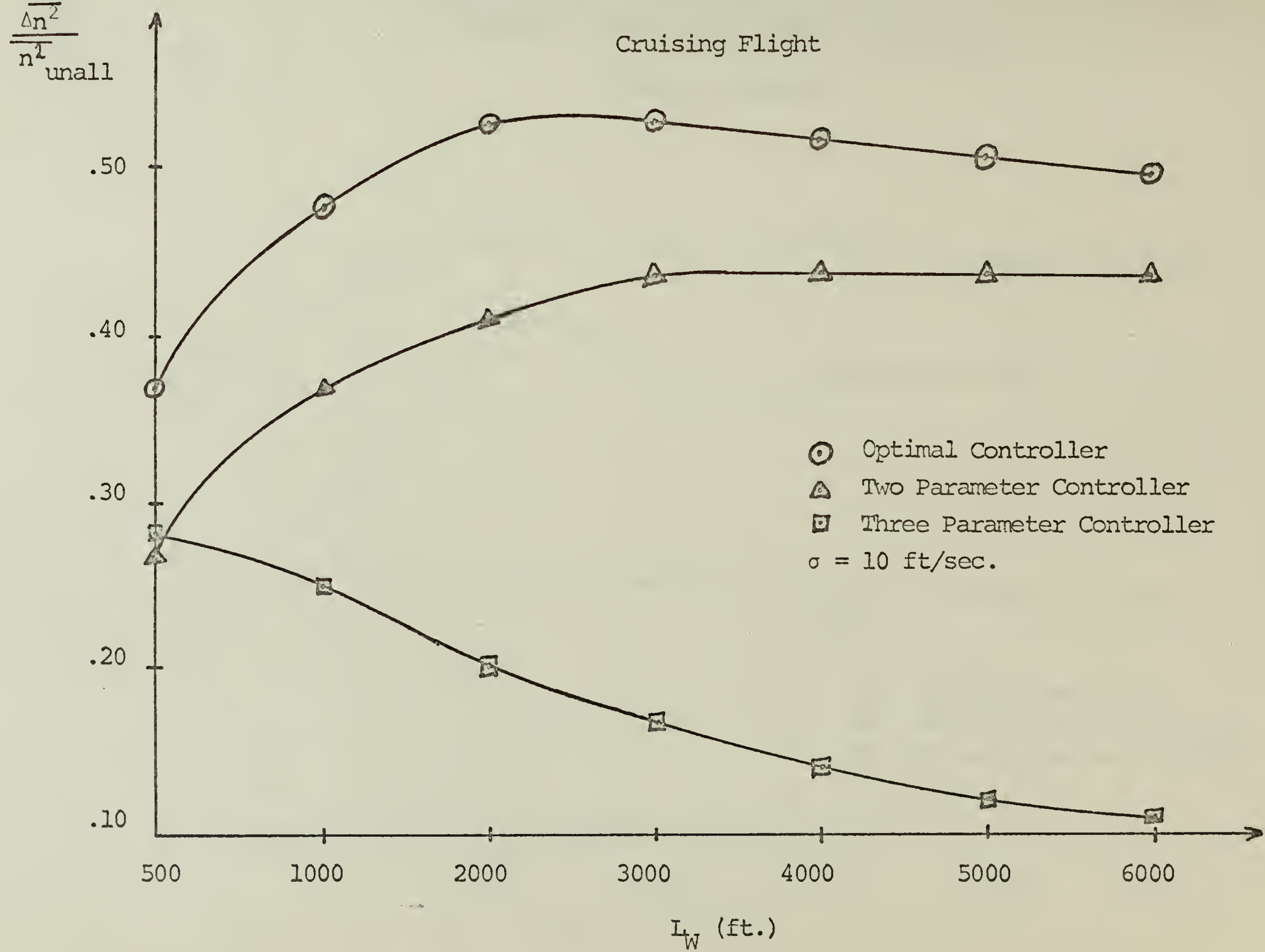


FIGURE 6. PERFORMANCE INDEX COMPARISONS

Landing Approach

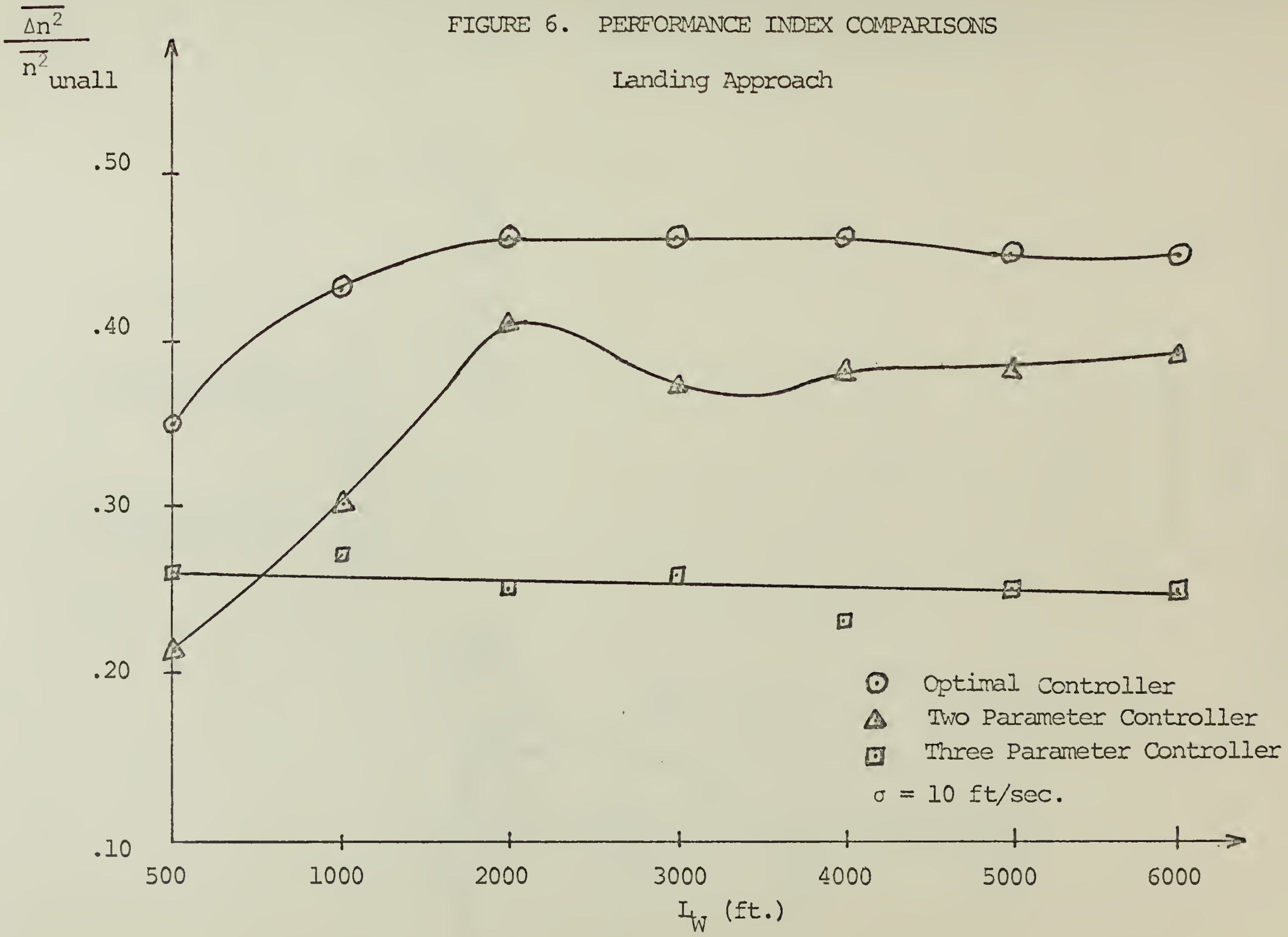


FIGURE 7

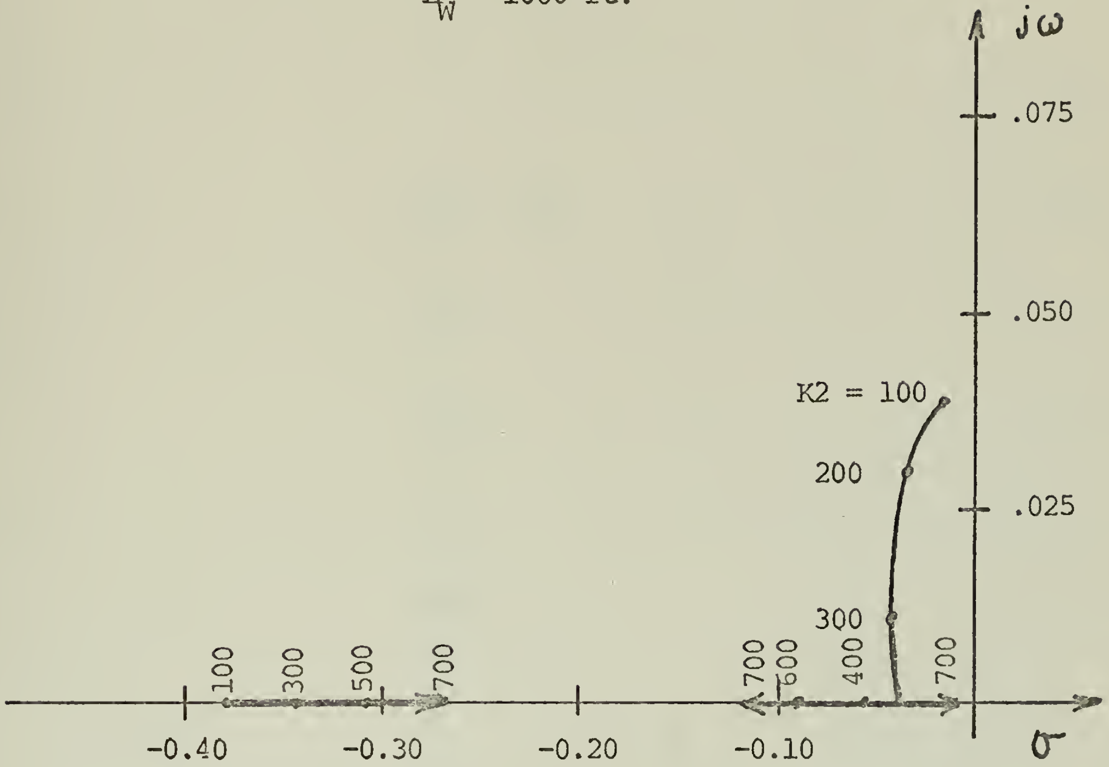
Three Parameter Gust Alleviation System

ROOT LOCUS

CRUISING FLIGHT

K1 = .785, K3 = -1.13

L_W = 1000 ft.



PERFORMANCE INDEX	
K2	P.I.
100	.0336
200	.0330
300	.0327
400	.0326
500	.0325
600	.0325
700	.0325

FIGURE 8

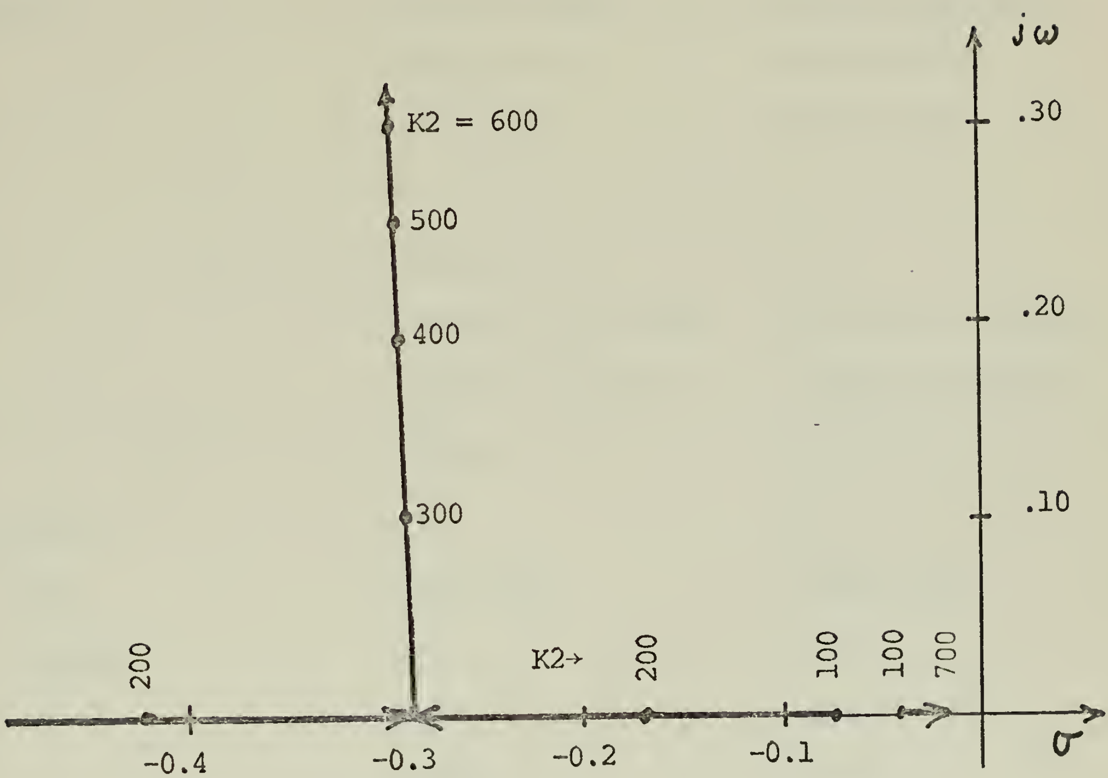
Three Parameter Gust Alleviation System

ROOT LOCUS

LANDING APPROACH

K1 = .785, K3 = -1.13

L_W = 1000 ft.



PERFORMANCE INDEX	
K2	P.I.
100	.0237
200	.0230
300	.0228
400	.0227
500	.0228
600	.0229
700	.0231

APPENDIX A

Aircraft Characteristics

The aircraft utilized in this report was the large, rigid, jet transport used by Etkin to illustrate application of theory he presented in Chapter 6 of Ref. 4. The cruising flight data were obtained from Ref. 4 and the landing approach data were obtained from Ref. 7.

Parameter	Cruising Flight	Landing Approach
W	1000,000 lbs.	1000,000 lbs.
S	1667 sq. ft.	1667 sq. ft.
A	7	7
Alt.	30,000 ft.	0 ft.
U_o	500 mph. or 733 fps.	200 mph. or 294 fps.
ρ	0.000889 slugs/ft.^{3}	0.002377 slugs/ft.^{3}
$\bar{c}/2$	7.7 ft.	7.7 ft.
$\mu = m/\rho s\bar{c}/2$	272	102
$t^* = \bar{c}/2U_o$	0.0105 sed.	0.0262 sec.
$i_B = B/\rho s\bar{c}/2$	1900	710
C_{L_o}	0.25	0.59
C_{D_o}	0.0188	0.0295
C_{z_α}	-4.9	-4.8
$C_{z_{\dot{\alpha}}}$	0	0
C_{m_α}	-0.488	-0.478
$C_{m_{\dot{\alpha}}}$	-4.2	-4.2

Aircraft Characteristics (Continued)

Parameter	Cruising Flight	Landing Approach
C_{m_q}	-22.9	-22.9
C_{z_q}	0	0
C_{z_η}	-0.24	-0.24
C_{m_η}	-0.72	-0.72
$C_{m_{\dot{\eta}}}$	0	0

APPENDIX B

Random Gust Fields and Their Use in Autopilot Design

Aircraft controllers designed to behave well when disturbed by sinusoidal, step or impulsive gust disturbances often perform poorly when subjected to actual atmospheric conditions. This has led to seeking better methods of defining the turbulence or gust fields. It has been discovered that if the atmospheric disturbance was treated as a random process and the aircraft controller designed using statistical concepts many of the above difficulties could be eliminated. Controllers could then be analytically designed, and built, which performed well when subjected to actual atmospheric conditions.

Statistical design techniques brought with them many new terms and concepts not used in the era of the so-called classical autopilot design. Terms such as stationary, homogeneous, isotropic, frozen turbulence, mean-square error, correlation functions and power spectral densities are now included in the analytical designer's vocabulary. This writer will not attempt to provide any mathematical derivations of these concepts and will not even discuss them at length. The interested reader is referred to the bibliography found in Ref. 4, pp. 339-340. A very readable review of this subject is found in Ref. 1, Chapter 9.

In order to utilize statistical techniques in gust alleviation design an expression for the power spectral density of the atmospheric turbulence is needed. Such expressions have been developed which show good agreement with spectra measured in cumulus clouds, thunderstorms and clear air. The spectra used for the random vertical gust field of

this report is found in Ref. 3, given by:

$$\Phi_{wg}(\Omega) = 2\sigma_w^2 L_w \frac{1}{1 + (L_w \Omega)^2} \tag{1}$$

where σ_w^2 is the mean square value of the vertical gust velocity. L_w is the characteristic gust length, and Ω is the spatial frequency. Equation (1) can be written in non-dimensional form as:

$$\Phi_{wg}(\hat{\omega}) = \frac{\sigma_w^2 \bar{c}^2}{L_w} \frac{1}{(\hat{\omega}^2 + (\bar{c}^2/4L_w^2))} \tag{2}$$

Although this expression is not as comprehensive as many expressions found for Φ_{wg}, it is used because it is easy to manipulate and it agrees fairly well with measured atmospheric power spectra. A form of Equation (1) was originally suggested in Ref. 8. σ_w has been found to vary from 0 to 40 feet per second and L_w from 500 to 6000 feet.

The power spectral density of any of the aircraft longitudinal motion variables such as angle of attack (α) can be obtained:

$$\Phi_{\alpha\alpha}(\hat{\omega}) = \Phi_{wg}(\hat{\omega}) \left| \frac{\alpha}{w_g}(\hat{\omega}) \right|^2 \tag{3}$$

Here $\frac{\alpha}{w_g}(j\hat{\omega})$ represents an aerodynamic transfer function, with $s = j\hat{\omega}$, and relates angle of attack and gust vertical velocity.

In addition, mean square values of longitudinal output variables can be obtained as:

$$\overline{\alpha^2} = \frac{1}{2\pi} \int_{\infty}^{\infty} \Phi_{\alpha\alpha}(\hat{\omega})d\hat{\omega} \tag{4}$$

$$= \frac{1}{2\pi} \int_{\infty}^{\infty} \Phi_{w_g}(\hat{\omega}) \left|\frac{\alpha}{w_g}(j\hat{\omega})\right|^2 d\hat{\omega} \tag{5}$$

Equations (3) and (5) provide the necessary tools for simple analysis and parameter optimization studies.

APPENDIX C

Determination of Transfer Functions

$$\frac{\dot{n}}{\alpha_g}(j\hat{\omega}) \text{ and } \frac{\eta}{\alpha_g}(j\hat{\omega})$$

I. Longitudinal Equations of Motion.

The aircraft longitudinal equations below were obtained from Ref. 4, page 133:

X force: $(2\mu D - 2C_{L_o}\tan\theta_o - C_{x_u})\hat{u} - C_{x_a}\alpha + C_{L_o}\theta = 0$

Z force: $(2C_{L_o} - C_{z_u})\hat{u} + (2\mu D - C_{z_{\dot{\alpha}}}D - C_{z_\alpha})\alpha - [(2\mu + C_{z_q})D - C_{L_o}\tan\theta_o]\theta - C_{z_\eta}\eta = 0$

Moment: $-C_{m_u}\hat{u} - (C_{m_{\dot{\alpha}}}D + C_{m_\alpha})\alpha + (i_B D^2 - C_{m_q}D)\theta - (C_{m_{\dot{\eta}}}D + C_{m_\eta})\eta = 0$

$\hat{q} - D\theta = 0 \qquad t^* = \ell/u_o \qquad \ell = \bar{c}/2$

Substituting $\alpha = \alpha + \alpha_g$, $\hat{q} = \hat{q} + \hat{q}_2$ and $\hat{p} = \hat{p} = \hat{p}_g$;

Neglecting changes in forward velocity and considering the X-force equation to be identically satisfied yields Equations (1) and (2) below. The equations are shown as they appear after the Laplace transform has been taken with zero initial condition assumed:

$$(2\mu s - C_{z_{\dot{\alpha}}}s - C_{z_\alpha})\alpha - (2\mu + C_{z_q})\hat{q} - C_{z_\eta}\eta = [C_{z_{\dot{\alpha}}}s + C_{z_\alpha} - C_{z_q}s]\alpha_g \quad (1)$$

$$-(C_{m_{\dot{\alpha}}}s + C_{m_\alpha})\alpha + (i_B s - C_{m_q})\hat{q} - [C_{m_{\dot{\eta}}}s + C_{m_\eta}]\eta \quad (2)$$

$$= [C_{m_{\dot{\alpha}}}s + C_{m_\alpha} - C_{m_q}s]\alpha_g$$

note: (a) $\hat{q} + \hat{q}_g = D\theta$

$$\hat{q}_g = \frac{d(\omega_g)dt}{dt \quad dx} = \frac{d \frac{\omega_g}{U_o}}{dt}$$

$$\hat{q}_g = -D\alpha_g$$

(b) The α_g terms above are included on the right side of Equations (1) and (2) to indicate their use as forcing functions.

II. Acceleration Equation.

The equation for the linear acceleration of the aircraft center of mass in the z direction is obtained from:

$$a_{c_z} = \dot{W} + PV - QU$$

by neglecting velocities in the y direction, assuming small disturbances, and using $\omega = u_o \sin\alpha$ Equation (3) is obtained:

$$a_{c_z} = \frac{U_o}{t^*}(D\alpha - \hat{q}) \qquad (3)$$

Now one can define the normal acceleration factor:

$$n(\hat{t}) = \frac{-a_{c_z}}{g} = \frac{2u_o^2}{2\bar{c}}(\hat{q} - D\alpha)$$

or in Laplace form

$$n(s) = \frac{2U_o^2}{2\bar{c}}(\hat{q} - s\alpha) \qquad (4)$$

III. Mathematical Model of Longitudinal Control System.

The equation for the longitudinal control system was obtained from the block diagram of the system and is given by:

$$K1\alpha + K2\hat{q} + (K3 - Ts - 1)\eta = 0 \qquad (5)$$

Note that the elevator-serro combination is approximated by a first order system with a time constant (T) of 1/10 second. T in the non-dimensionalized form is given by:

$$T^* = T/t^* = \frac{2TU_o}{\bar{c}}$$

IV. Simultaneous Solution.

From Equations (1), (2), (4), and (5) a system of equations is obtained of the form:

$$\begin{aligned}
A_{11}\alpha + A_{12}\hat{q} + A_{13}\eta + A_{14}n &= b_1\alpha_g \\
A_{21}\alpha + A_{22}\hat{q} + A_{23}\eta + A_{24}n &= b_2\alpha_g \\
A_{31}\alpha + A_{32}\hat{q} + A_{33}\eta + A_{34}n &= b_3\alpha_g \\
A_{41}\alpha + A_{42}\hat{q} + A_{43}\eta + A_{44}n &= b_4\alpha_g
\end{aligned} \qquad (6)$$

where

$A_{11} = 2\mu s - C_{z_{\dot{\alpha}}} s - C_{z_\alpha}$

$A_{12} = -2\mu - C_{z_q}$

$A_{13} = - C_{z_\eta}$

$A_{14} = 0$

$b_1 = C_{z_{\dot{\alpha}}} s + C_{z_\alpha} - C_{z_q} s$

$A_{21} = C_{m_{\dot{\alpha}}} s - C_{m_\alpha}$

$A_{22} = i_B s - C_{m_q}$

$A_{23} = -C_{m_{\dot{\eta}}} - C_{m_\eta}$

$A_{24} = 0$

$b_2 = C_{m_{\dot{\alpha}}} s + C_{m_\alpha} - C_{m_q} s$

$A_{31} = K1$

$A_{32} = K2$

$A_{33} = K3 - T^*s - 1$

$A_{34} = 0$

$b_3 = 0$

$A_{41} = -2U_o^2 s/g\bar{c}$

$A_{42} = 2U_o^2/g\bar{c}$

$A_{43} = 0$

$A_{44} = -1$

$b_4 = 0$

The transfer function $\bar{n}(s)/\alpha_g(s)\big|_{s=j\hat{\omega}}$ is given by:

$$\frac{\bar{n}}{\alpha_g}(s)\Bigg|_{s=j\hat{\omega}} = \frac{\begin{vmatrix} A_{11} & A_{12} & A_{13} & b_1 \\ A_{21} & A_{22} & A_{23} & b_2 \\ A_{31} & A_{32} & A_{33} & b_3 \\ A_{41} & A_{42} & A_{43} & b_4 \end{vmatrix}}{\begin{vmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{vmatrix}} \qquad (7)$$

$\bar{\eta}(s)/\alpha_g(s)\big|_{s=j\hat{\omega}}$ is obtained in a similar manner.

APPENDIX D

Computer Program

I. The computer program was written to provide a solution to a performance index given by:

$$I_n = \frac{1}{2\pi}\int_{\infty}^{\infty} \Phi_{\alpha g} \left|\frac{n}{\alpha_g}(j\hat{\omega})\right|^2 d\hat{\omega} + \frac{1}{2\pi}\int_{\infty}^{\infty} \Phi_{\alpha_g}(\hat{\omega}) \left|\frac{n}{\alpha_g}(j\hat{\omega})\right|^2 d\hat{\omega} \qquad (1)$$

where

$$I_{\alpha g}(\hat{\omega}) = \frac{\sigma^2 \bar{c}}{L_W U_O^2} \frac{1}{\hat{\omega}^2 + \frac{\bar{c}^2}{4L_W^{\,2}}} \qquad (2)$$

$$\left|\frac{n}{\alpha_g}(j\hat{\omega})\right|^2 = \left|\frac{x1(j\hat{\omega})^2 + x2(j\hat{\omega}) + x3}{x4(j\hat{\omega})^3 + x5(j\hat{\omega})^2 + x6(j\hat{\omega}) + x7}\right| \qquad (3)$$

$$\left|\frac{n}{\alpha_g}(j\hat{\omega})\right|^2 = \left|\frac{z1(j\hat{\omega})^3 + x2(j\hat{\omega})^2 + z3(j\hat{\omega}) + z4}{x4(j\hat{\omega})^3 + x5(j\hat{\omega})^2 + x6(j\hat{\omega}) + x7}\right| \qquad (4)$$

In Equations (3) and (4), x1 through x7 and z1 through z4 contain the variables K1, K2, and K3 and coefficients of these variables. The coefficients are functions of the aircraft stability derivatives and the gust characteristic length.

It can be seen that the performance index can be written in the form:

$$I_n = \frac{1}{2\pi}\int_{\infty}^{\infty} \frac{C(j\hat{\omega})C(-j\hat{\omega})}{d(j\hat{\omega})d(-j\hat{\omega})}\, d\omega \qquad (5)$$

This integral is evaluated for n = 1 through n = 10 in Ref. 5. For the performance index of this problem $I_n = I_4$ where

$$I_4 = \frac{c_3^2(-d_0^2d_3+d_0d_1d_2)+(c_2^2-2c_1c_3)d_0d_1d_4+(c_1^2-2c_0c_2)d_0d_3d_4+c_0^2(-d_1d_4^2+d_2d_3d_4)}{2d_0d_4(-d_0d_3^2-d_1^2d_4+d_1d_2d_3)} \tag{6}$$

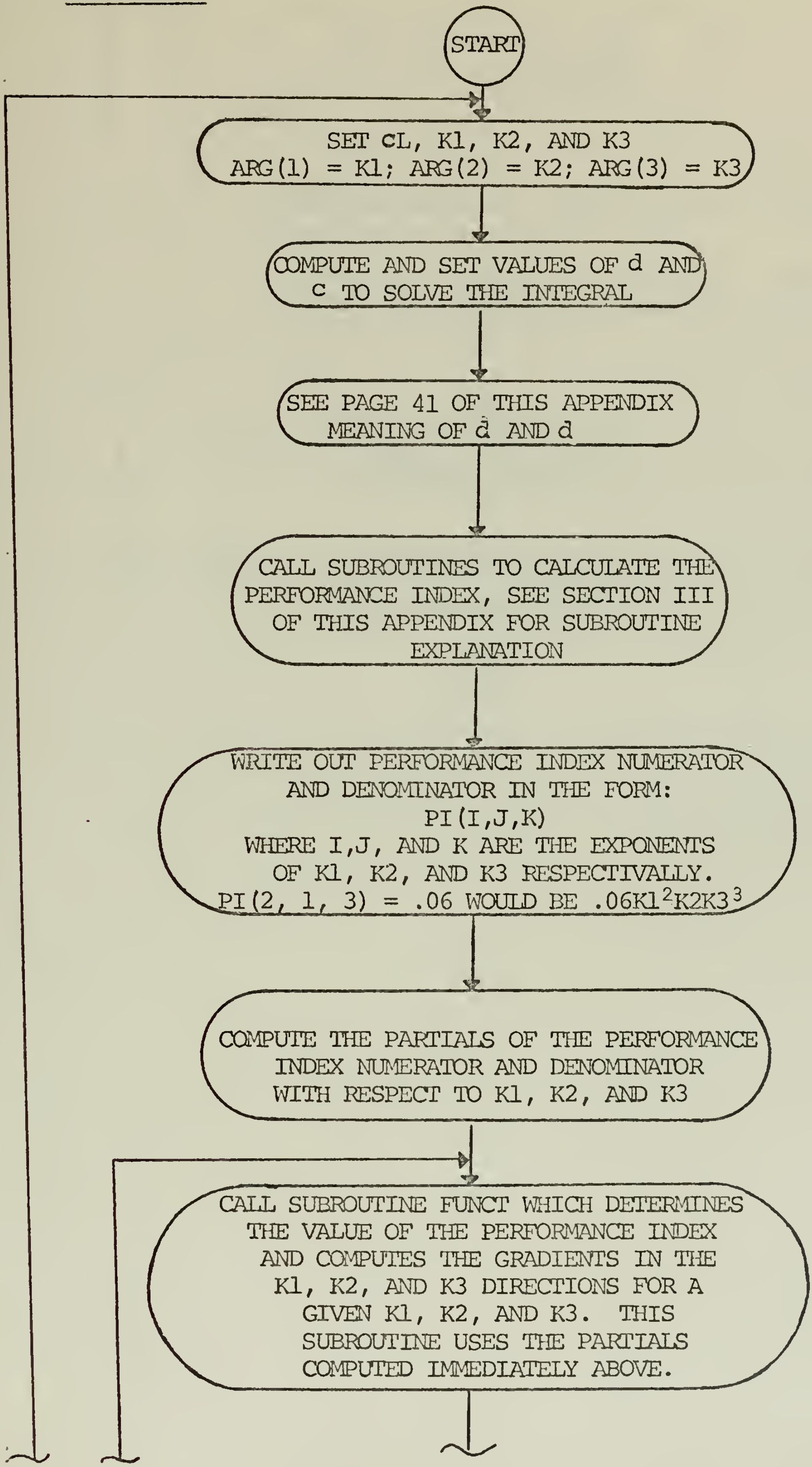
$$c(j\hat{\omega}) + c_3(j\hat{\omega})^3 + c_2(j\hat{\omega})^2 + c_1(j\hat{\omega}) + c_0$$

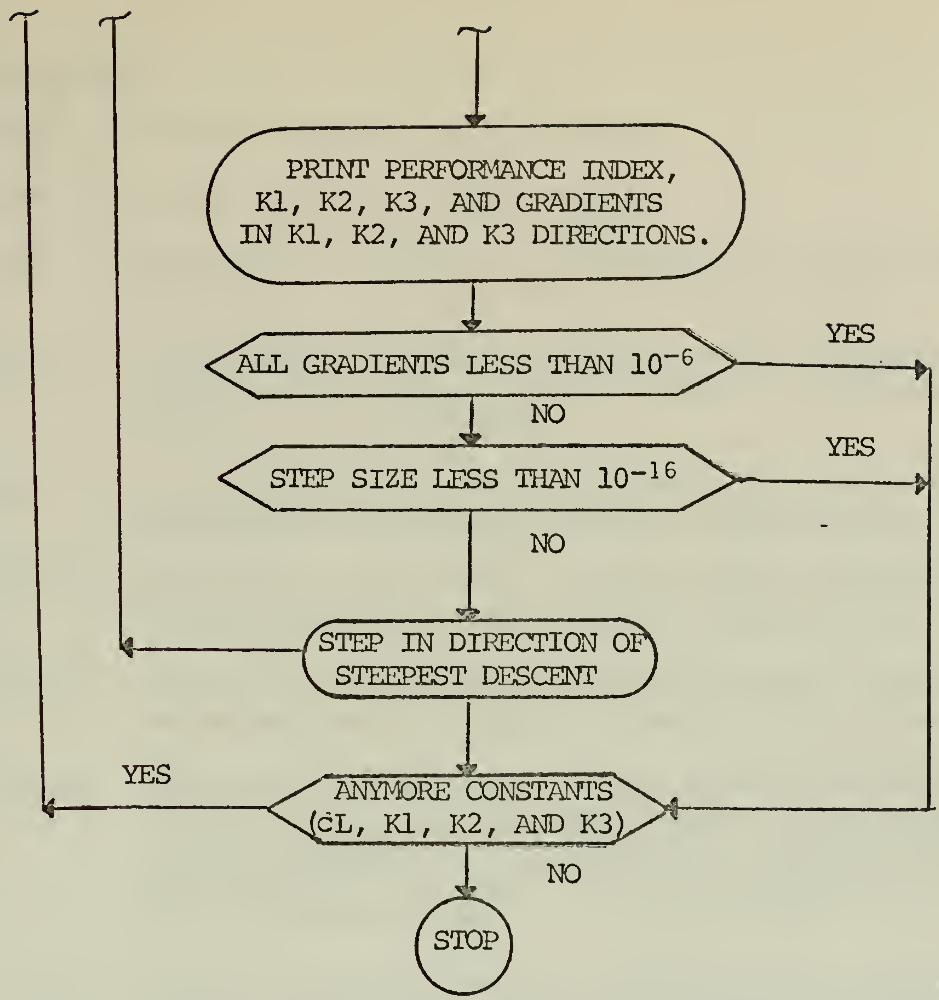
$$d(j\hat{\omega}) = d_4(j\hat{\omega})^4 + d_3(j\hat{\omega})^3 + d_2(j\hat{\omega})^2 + d_1(j\hat{\omega}) + d_0$$

The I_4 equation was programmed and solved on the digital computer. The program was written so that the type aircraft could be changed, i.e. all stability derivatives were retained in the computation of x1 through x7 and z1 through z4. The gust characteristic length (CL) was set at the beginning of the program. To facilitate handling of the variables K1, K2, and K3 their coefficients were stored in 3 dimensional arrays according to the exponents of the variables. For example, the coefficient of $K1^2K2K3^3$ would be stored in an array at location (3,2,4). The value stored in this location was printed out as PI(2,1,3) = (value stored). The arrays were manipulated to solve for the performance index in terms of K1, K2, and K3.

A minimum value of PI(K1, K2, K3) was determined by a steepest descent method. A starting point for the minimization procedure was obtained by computing values of the performance index as K1, K2, and K3 varied over a large range. The point where the performance index was a minimum was then used as a starting point for the steepest descent method. Several other starting points were utilized to help ensure that a proper minimum was reached.

II. Flow Chart




PRINT PERFORMANCE INDEX, K1, K2, K3, AND GRADIENTS IN K1, K2, AND K3 DIRECTIONS.
ALL GRADIENTS LESS THAN 10-6
YES
NO
STEP SIZE LESS THAN 10-16
YES
NO
STEP IN DIRECTION OF STEEPEST DESCENT
YES
ANYMORE CONSTANTS (ĊL, K1, K2, AND K3)
NO
STOP

III. Subroutines

ZERO3 - Places zeros in a 3x3x3 matrix.

ZERO4 - Places zeros in a 4x4x4 matrix.

MULT1 - Multiplies a first order polynomial by a first order polynomial and stores answer in a 3x3x3 matrix.

MULT2 - Determines the product of a first order polynomial times a first order polynomial times 2.0 and stores answer in a 3x3x3 matrix.

MULT3 - Multiplies a first order polynomial by a second order polynomial and stores answer in a 4x4x4 matrix.

SUBT1 - Subtracts a third order polynomial from a third order polynomial and stores answer in a 4x4x4 matrix.

SUBT2 - Subtracts a second order polynomial from a second order polynomial and stores answer in a 4x4x4 matrix.

STUFFP - Determines the product of a second order polynomial times a third order polynomial times a constant and stores the answer in a 6x6x6 matrix. Separate 6x6x6 matrices hold the numerator and the denominator of the performance index.

FUNCT - Takes the performance index numerator and denominator arrays, the specific values of K1, K2, and K3, and the arrays holding the partials of the performance index numerator and denominator and calculates the value of the performance index and the gradients in the K1, K2 and K3 directions.

```
// EXEC FORTCLG,REGION.GO=130K
//FORT.SYSIN DD *
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PIN(6,6,6), PID(6,6,6), PIL(6,6,6)
      DIMENSION DO(2,2,2), D1(2,2,2), D2(2,2,2), D3(2,2,2), D4(2,2,2)
      DIMENSION CO(2,2,2), C1(2,2,2), C2(2,2,2), C3(2,2,2)
      DIMENSION HOLD31(3,3,3), HOLD32(3,3,3)
      DIMENSION HOLD41(4,4,4), HOLD42(4,4,4)
      DIMENSION PRTLK3(6,6,6), PRTLK2(6,6,6), PRTLK1(6,6,6)
      DIMENSION PRLDK3(6,6,6), PRLDK2(6,6,6), PRLDK1(6,6,6)
      DIMENSION ARG(3), GRAD(3)
      ILOOP=-1
C
C    THIS PROGRAM SOLVES:
C
C              +INFINITY
C               1         C(JW)C(-JW)
C       I   =---- | DW  -----------
C        N    2PI |       D(JW)D(-JW)
C              -INFINITY
C
C    FOR N=4.  REFERENCE - NEWTON, G.C. JR, AND OTHERS, "ANALYTICAL DESIGN
C    OF LINEAR FEEDBACK CONTROLS", P.372, JOHN WILEY & SONS, 1957.
C    IF DESIRED, THE PROGRAM WILL LOOK FOR THE VALUES OF K1, K2, AND K3
C    WHICH MAKES I A MINIMUM VALUE BY A METHOD OF STEEPEST DESCENT.
C    IN THIS PROGRAM I REFERRED TO AS PI (PERFORMANCE INDEX).
C
C    PROGRAM PARAMETERS
C
C      ITPFLT - ITPFLT=1   INCLUDES ONLY CRUISING STABILITY DERIVATIVES
C               ITPFLT=0   INCLUDES LANDING APPROACH DERIVATIVES ONLY
C      IMIN -   IMIN=1     INCLUDES ROUTINE TO MINIMIZE PI
C               IMIN=0     SKIPS MINIMIZATION ROUTINE
C      IELRMS - IELRMS=1   INCLUDES THE ELEVATOR CONTRIBUTION TO THE PI
C               IELRMS=0 SKIPS THE ELEVATOR CONTRIBUTION
C      CL     - CL=#       PUT # 500.0, 1000.0, 2000.0, 3000.0, 4000.0,
C                          5000.0, OR 6000.0 AND PROGRAM WILL USE ONLY THAT
C                          CHARACTERISTIC GUST LENGTH
C               CL=0.0     PROGRAM WILL ITERATE THRU ALL 7 VALUES OF CL
C
C    SET ITPFLT, IMIN, IELRMS, AND CL BELOW.
C
      CL=6000.0
      ITPFLT=1
      IMIN=1
      IELRMS=1
C
      IF(ITPFLT.EQ.1)GO TO 1
```

```
C
C **** CONSTANTS FOR THE LANDING APPROACH ******************
      WRITE(6,3)
    3 FORMAT(' ','LANDING APPROACH')
      UZERO=294.0
      T=0.1
      GRAV=32.2
      SIGMA=10.0
      CBAR=15.4
      CZAD=0.0
      CZQ=0.0
      CIB=710.0
      CZA=-4.8
      CMQ=-22.9
      CMU=102.0
      CZN=-0.24
      CMAD=-4.2
      CMND=0.0
      CMN=-0.72
      CMA=-.478
C **** END OF CONSTANTS FOR LANDING APPROACH **************
C
      GO TO 6
C
C **** CONSTANTS FOR CRUISING FLIGHT ************************
    1 WRITE(6,4)
    4 FORMAT(' ','CRUISING FLIGHT')
      UZERO=733.0
      T=0.1
      GRAV=32.2
      SIGMA=10.0
      CBAR=15.4
      CZAD=0.0
      CZQ=0.0
      CIB=1900.0
      CZA=-4.9
      CMQ=-22.9
      CMU=272.0
      CZN=-0.24
      CMAD=-4.2
      CMND=0.0
      CMN=-0.72
      CMA=-0.488
C **** END OF CONSTANTS FOR CRUISING FLIGHT *******************
C
    6 CONTINUE
C
C   CHECK CL AND SET UP PROGRAM FOR VARYING CL OR FOR MAKING ONLY
```

```
C     ONE PASS THRU PROGRAM WITH A SPECIFIC CL.
C
      CLH=CL
      IF(CL.EQ.500.0)GO TO 800
      IF(CL.EQ.0.0)GO TO 800
      ICLPIC=IFIX(SNGL(CL/1000.0))
      ILOOP=ILOOP+1+ICLPIC
      GO TO(801,802,803,804,805,806),ICLPIC
    8 IF(CLH.NE.0.0)ILOOP=7
      DO 10 I=1,5
   10 WRITE(6,11)
   11 FORMAT('0')
      WRITE(6,5)CL
    5 FORMAT('0','CHARACTERISTIC LENGTH =',F7.1)
C
C     ZERO ARRAYS
C
      DO 100 I=1,2
      DO 110 J=1,2
      DO 120 K=1,2
      D0(I,J,K)=0.0
      D1(I,J,K)=0.0
      D2(I,J,K)=0.0
      D3(I,J,K)=0.0
      D4(I,J,K)=0.0
      C0(I,J,K)=0.0
      C1(I,J,K)=0.0
      C2(I,J,K)=0.0
      C3(I,J,K)=0.0
  120 CONTINUE
  110 CONTINUE
  100 CONTINUE
      DO 130 I=1,6
      DO 140 J=1,6
      DO 150 K=1,6
      PIN(I,J,K)=0.0
      PIL(I,J,K)=0.0
      PID(I,J,K)=0.0
      PRTLK3(I,J,K)=0.0
      PRTLK2(I,J,K)=0.0
      PRTLK1(I,J,K)=0.0
      PRLDK3(I,J,K)=0.0
      PRLDK2(I,J,K)=0.0
      PRLDK1(I,J,K)=0.0
  150 CONTINUE
  140 CONTINUE
  130 CONTINUE
```

```
C
C     **********   PROGRAM CONSTANTS   ***************
C
      A=SIGMA*SIGMA*CBAR/(CL*UZERO*UZERO)
      Y=CBAR/(2.0*CL)
      TIMND=2.0*T*UZERO/CBAR

      X1K100=(CZAD-CZQ)*CIB
      X2K100=CZA*CIB-CZQ*CMQ+CZAD*CMQ
      X3K100=CZA*CMQ
      X4K000=CIB*TIMND*(2.0*CMU-CZAD)
      X5K001=CIB*(CZAD-2.0*CMU)
      X5K010=-2.0*CMU*CMND
      X5K000=CIB*(2.0*CMU-CZAD)-TIMND*(CZA*CIB+2.0*CMU*CMQ-CZAD*CMQ)
      X5K000=X5K000-TIMND*(CMAD+2.0*CMU*CMAD+CZQ*CMAD)
      X6K001=CZA*CIB+2.0*CMU*CMQ-CZAD*CMQ
      X6K001=X6K001+CMAD*(2.0*CMU+CZQ)
      X6K100=-CZN*CIB
      X6K100=X6K100-CMND*(2.0*CMU+CZQ)
      X6K010=CZAD*CMND+CZA*CMND-CZN*CMAD-2.0*CMU*CMN
      X6K000=-CMQ*(2.0*CMU-CZA*TIMND-CZAD)-CZA*CIB
      X6K000=X6K000-TIMND*(2.0*CMU*CMA+CZQ*CMA)-CMAD*(2.0*CMU+CZQ)
      X7K001=-CZA*CMQ
      X7K001=X7K001+CMA*(2.0*CMU+CZQ)
      X7K010=CMN*(CZAD+CZA)-CZN*CMA
      X7K100=+CZN*CMQ
      X7K100=X7K100-2.0*CMU*CMN-CZQ*CMND
      X7K000=+CZA*CMQ
      X7K000=X7K000-CMA*(2.0*CMU+CZQ)
      ACCFAC=2.0*UZERO*UZERO/(GRAV*CBAR)
      Z1K001=ACCFAC*(CIB*(CZAD-CZQ))
      Z1K010=ACCFAC*(CMND*(CZAD-CZQ))
      Z1K000=-ACCFAC*TIMND*(2.0*CMU*(CMQ-CMAD)+CZAD*CMAD-CZAD*CMQ)
      Z1K000=Z1K000+ACCFAC*(CMAD*TIMND*(-2.0*CMU-CZQ+CZAD-CZQ))
      Z1K000=Z1K000+ACCFAC*(CMQ*TIMND*(2.0*CMU+CZAD))
      Z1K000=Z1K000+ACCFAC*(CIB*(CZQ-CZAD-CZA*TIMND))
      Z2K001=-ACCFAC*(2.0*CMU*CMAD-CZAD*CMAD-2.0*CMU*CMQ+CZAD*CMQ)
      Z2K001=Z2K001+ACCFAC*(CMAD*(2.0*CZQ+2.0*CMU-CZAD)+CMQ*(-2.0*CMU-CZ
     1AD)+CZA*CIB)
      Z2K010=ACCFAC*(CZN*(CMQ-CMAD)+CMN*(CZAD-CZQ)+CZA*CMND)
      Z2K100=ACCFAC*(CMND*(CZAD-CZQ))
      Z2K000=-ACCFAC*(TIMND*(CZA*CMAD-2.0*CMU*CMA+CZAD*CMA-CZA*CMQ)+(CMA
     1D*CZAD-2.0*CMU*CMAD+2.0*CMU*CMQ-CZAD*CMQ))
      Z2K000=Z2K000+ACCFAC*(CMAD*(-2.0*CMU-2.0*CZQ+CZAD+CZA*TIMND))
      Z2K000=Z2K000+ACCFAC*(CMA*TIMND*(CZAD-2.0*CZQ-2.0*CMU))
      Z2K000=Z2K000+ACCFAC*(CMQ*(2.0*CMU+CZQ+CZAD+CZA*TIMND-CZQ))
      Z2K000=Z2K000-ACCFAC*CZA*CIB
      Z3K001=2.0*ACCFAC*(CMA*CZQ-CZA*CMQ)
```

```
      Z3K010=ACCFAC*(CZA*CMN-CMA*CZN)
      Z3K100=ACCFAC*(CZN*CMQ-CZN*CMAD)
      Z3K100=Z3K100+ACCFAC*(CZAD*CMN-CZQ*CMN+CZA*CMND)
      Z3K000=2.0*ACCFAC*(CZA*CMQ-CMA*CZQ)
      Z4K100=-ACCFAC*CZN*CMA
      Z4K100=Z4K100+ACCFAC*CZA*CMN
C
C    SET VALUES OF D AND C FOR COMPUTATION OF THE DENOMINATOR OF THE PI AND
C    THE NUMERATOR OF THE ACCELERATION CONTRIBUTION TO THE PI.
C
      D4(1,1,1)=X4K000
      D3(1,1,2)=X5K001
      D3(1,2,1)=X5K010
      D3(1,1,1)=X5K000+X4K000*Y
      D2(1,1,2)=X6K001+X5K001*Y
      D2(1,2,1)=X6K010+X5K010*Y
      D2(2,1,1)=X6K100
      D2(1,1,1)=X6K000+X5K000*Y
      D1(1,1,2)=X7K001+X6K001*Y
      D1(1,2,1)=X7K010+X6K010*Y
      D1(2,1,1)=X7K100+X6K100*Y
      D1(1,1,1)=X7K000+X6K000*Y
      D0(1,1,2)=X7K001*Y
      D0(2,1,1)=X7K100*Y
      D0(1,1,1)=X7K000*Y
      C0(2,1,1)=Z4K100
      C1(1,1,1)=Z3K000
      C1(2,1,1)=Z3K100
      C1(1,2,1)=Z3K010
      C1(1,1,2)=Z3K001
      C2(1,1,1)=Z2K000
      C2(1,2,1)=Z2K010
      C2(1,1,2)=Z2K001
      C2(2,1,1)=Z2K100
      C3(1,1,1)=Z1K000
      C3(1,1,2)=Z1K001
      C3(1,2,1)=Z1K010
      JFLAG=0
C
C    ROUTINE TO CALCULATE PI STARTS HERE AND ENDS AT STATEMENT 370
C
      CALL ZERO3(HOLD31)
      CALL ZERO4(HOLD41)
      CALL MULT1(HOLD31,D0,D0)
      CALL MULT3(HOLD41,D3,HOLD31)
      CALL ZERO3(HOLD31)
      CALL MULT1(HOLD31,D1,D2)
      CALL ZERO4(HOLD42)
```

```
      CALL  MULT3(HOLD42,D0,HOLD31)
      CALL  SUBT1(HOLD42,HOLD41)
      CALL  ZERO3(HOLD31)
      CALL  MULT1(HOLD31,C3,C3)
      CALL  STUFFP(PIN,HOLD31,HOLD42,A)
  160 CALL  ZERO3(HOLD31)
      CALL  MULT1(HOLD31,C2,C2)
      CALL  ZERO3(HOLD32)
      CALL  MULT2(HOLD32,C1,C3)
      CALL  SUBT2(HOLD31,HOLD32)
      CALL  ZERO3(HOLD32)
      CALL  MULT1(HOLD32,D0,D1)
      CALL  ZERO4(HOLD41)
      CALL  MULT3(HOLD41,D4,HOLD32)
      CALL  STUFFP(PIN,HOLD31,HOLD41,A)
      IF(JFLAG.EQ.1)CALL  STUFFP(PIL,HOLD31,HOLD41,A)
      CALL  ZERO3(HOLD31)
      CALL  MULT1(HOLD31,C1,C1)
      CALL  ZERO3(HOLD32)
      CALL  MULT2(HOLD32,C0,C2)
      CALL  SUBT2(HOLD31,HOLD32)
      CALL  ZERO3(HOLD32)
      CALL  MULT1(HOLD32,D0,D3)
      CALL  ZERO4(HOLD41)
      CALL  MULT3(HOLD41,D4,HOLD32)
      CALL  STUFFP(PIN,HOLD31,HOLD41,A)
      IF(JFLAG.EQ.1)CALL  STUFFP(PIL,HOLD31,HOLD41,A)
      CALL  ZERO3(HOLD31)
      CALL  MULT1(HOLD31,C0,C0)
      CALL  ZERO3(HOLD32)
      CALL  MULT1(HOLD32,D2,D3)
      CALL  ZERO4(HOLD41)
      CALL  MULT3(HOLD41,D4,HOLD32)
      CALL  ZERO3(HOLD32)
      CALL  MULT1(HOLD32,D1,D4)
      CALL  ZERO4(HOLD42)
      CALL  MULT3(HOLD42,D4,HOLD32)
      CALL  SUBT1(HOLD41,HOLD42)
      CALL  STUFFP(PIN,HOLD31,HOLD41,A)
      IF(JFLAG.EQ.1)CALL  STUFFP(PIL,HOLD31,HOLD41,A)
      IF(JFLAG.EQ.1)GO TO 300
      DO 170 I=1,6
      DO 180 J=1,6
      DO 190 K=1,6
      L=I-1
      M=J-1
      N=K-1
      IF(PIN(I,J,K).EQ.0.0)GO TO 190
```

```
      WRITE(6,195)L,M,N,PIN(I,J,K)
  195 FORMAT(' ','ACCELERATION NUMERATOR(',3I1,')=',D16.6)
  190 CONTINUE
  180 CONTINUE
  170 CONTINUE
C
C     SKIP ELEVATOR CONTRIBUTION TO PI IF DESIRED.
C
      IF(IELRMS.EQ.0)GO TO 336
C
C     ZERO ARRAYS.
C
  196 DO 200 I=1,6
      DO 210 J=1,6
      DO 220 K=1,6
      C0(I,J,K)=0.0
      C1(I,J,K)=0.0
      C2(I,J,K)=0.0
      C3(I,J,K)=0.0
  220 CONTINUE
  210 CONTINUE
  200 CONTINUE
C
C     THE PORTION OF THE PROGRAM BETWEEN HERE AND THE NEXT COMMENT CARD
C     COMPUTES THE CONTRIBUTION OF THE ELEVATOR DEFLECTION TO THE PI.
C
      Z2K010=2.0*CMU*(CMAD-CMQ)+CZAD*CMQ-CMAD*CZQ
      Z2K100=CIB*(CZAD-CZQ)
      Z3K010=2.0*CMU*CMA+CZA*CMQ-CMA*CZQ
      Z3K100=2.0*CMU*(CMAD-CMQ)+CZA*CIB+CZQ*CMAD-CZAD*CMQ
      Z4K100=2.0*CMU*CMA+CZQ*CMA-CZA*CMQ
      C2(1,2,1)=Z2K010
      C2(2,1,1)=Z2K100
      C1(1,2,1)=Z3K010
      C1(2,1,1)=Z3K100
      C0(2,1,1)=Z4K100
      JFLAG=JFLAG+1
      IF(JFLAG.EQ.1)GO TO 160
  300 DO 310 I=1,6
      DO 320 J=1,6
      DO 330 K=1,6
      L=I-1
      M=J-1
      N=K-1
      IF(PIL(I,J,K).EQ.0.0)GO TO 330
      WRITE(6,335)L,M,N,PIL(I,J,K)
  335 FORMAT(' ','ELEVATOR NUMERATOR(',3I1,')=',D16.6)
  330 CONTINUE
```

```
  320 CONTINUE
  319 CONTINUE
C
C **** ELEVATOR COMPUTATION ENDS HERE
C
  336 A=1.0
      CALL ZERO3(HOLD31)
      CALL MULT1(HOLD31,D1,D2)
      CALL ZERO4(HOLD41)
      CALL MULT3(HOLD41,D3,HOLD31)
      CALL ZERO3(HOLD31)
      CALL MULT1(HOLD31,D1,D1)
      CALL ZERO4(HOLD42)
      CALL MULT3(HOLD42,D4,HOLD31)
      CALL SUBT1(HOLD41,HOLD42)
      CALL ZERO3(HOLD31)
      CALL MULT1(HOLD31,D0,D3)
      CALL ZERO4(HOLD42)
      CALL MULT3(HOLD42,D3,HOLD31)
      CALL SUBT1(HOLD41,HOLD42)
      CALL ZERO3(HOLD31)
      CALL MULT2(HOLD31,D0,D4)
      CALL STUFFP(PID,HOLD31,HOLD41,A)
      DO 340 I=1,6
      DO 350 J=1,6
      DO 360 K=1,6
      L=I-1
      M=J-1
      N=K-1
      IF(PIN(I,J,K).EQ.0.0)GO TO 360
      WRITE(6,365)L,M,N,PIN(I,J,K)
  365 FORMAT(' ','PERFORMANCE INDEX NUMERATOR(',3I1,')=',D16.6)
  360 CONTINUE
  350 CONTINUE
  340 CONTINUE
      DO 370 I=1,6
      DO 380 J=1,6
      DO 390 K=1,6
      IF(PID(I,J,K).EQ.0.0)GO TO 390
      L=I-1
      M=J-1
      N=K-1
      WRITE(6,395)L,M,N,PID(I,J,K)
  395 FORMAT(' ','PERFORMANCE INDEX DENOMINATOR(',3I1,')=',D16.6)
  390 CONTINUE
  380 CONTINUE
  370 CONTINUE
C
```

```
C
C     START ROUTINE TO COMPUTE THE PARTIALS OF THE PI NUMERATOR AND
C     DENOMINATOR WITH RESPECT TO K1, K2, AND K3.
C     ROUTINE ENDS AT STATEMENT VALUE 700.
C
      DO 700 I=1,6
      DO 710 J=1,6
      DO 720 K=1,6
      IF((PIN(I,J,K).EQ.0.0).AND.(PID(I,J,K).EQ.0.0))GO TO 720
      L=I-1
      M=J-1
      N=K-1
      FL=DFLOAT(L)
      FM=DFLOAT(M)
      FN=DFLOAT(N)
      IF(N.EQ.0)GO TO 760
      IF(PIN(I,J,K).NE.0.0)PRTLK3(I,J,N)=FN*PIN(I,J,K)
      IF(PID(I,J,K).NE.0.0)PRLDK3(I,J,N)=FN*PID(I,J,K)
  760 IF(M.EQ.0)GO TO 750
      IF(PIN(I,J,K).NE.0.0)PRTLK2(I,M,K)=FM*PIN(I,J,K)
      IF(PID(I,J,K).NE.0.0)PRLDK2(I,M,K)=FM*PID(I,J,K)
  750 IF(L.EQ.0)GO TO 740
      IF(PIN(I,J,K).NE.0.0)PRTLK1(L,J,K)=FL*PIN(I,J,K)
      IF(PID(I,J,K).NE.0.0)PRLDK1(L,J,K)=FL*PID(I,J,K)
  740 CONTINUE
  720 CONTINUE
  710 CONTINUE
  700 CONTINUE
C
C     START ROUTINE TO MINIMIZE AND PRINT PERFORMANCE INDEX. ROUTINE ENDS
C     AT STATEMENT VALUE 800.
C
      STPSIZ=.5
      WRITE(6,780)
      WRITE(6,781)
  780 FORMAT('0',T4,'PERFORMANCE')
  781 FORMAT(' ',T4,'INDEX',T19,'K1',T34,'K2',T49,'K3',T64,'GRAD1',T79,
     C'GRAD2',T94,'GRAD3',T109,'TEST1',T121,'TEST2')
      CALL FUNCT(ARG,VAL,GRAD,PIN,PID,PRTLK1,PRTLK2,PRTLK3,PRLDK1,PRLDK2
     C,PRLDK3)
  795 CONTINUE
      WRITE(6,782)VAL,ARG(1),ARG(2),ARG(3),GRAD(1),GRAD(2),GRAD(3)
  782 FORMAT(' ',T4,D11.5,T19,D11.5,T34,D11.5,T49,D11.5,T64,D11.5,T79,D1
     C1.5,T94,D11.5)
C
C     SKIP MINIMIZATION PROCEDURE IF DESIRED.
C
      IF(IMIN.EQ.0)GO TO 800
      STEP=STPSIZ
```

```
      SARG2=ARG(2)
      SARG1=ARG(1)
      SARG3=ARG(3)
      SGRD1=GRAD(1)
      SGRD2=GRAD(2)
      SGRD3=GRAD(3)
      SVAL=VAL
      IF((DABS(GRAD(1)).LE.1D-6).AND.(DABS(GRAD(2)).LE.1D-6).AND.(DABS(G
     CRAD(3)).LE.1D-6))GO TO 789
      SUMSQ=GRAD(1)**2+GRAD(2)**2+GRAD(3)**2
      UNITGR=DSQRT(SUMSQ)
  786 DLTAK1=-STEP*SGRD1/UNITGR
      DLTAK2=-STEP*SGRD2/UNITGR
      DLTAK3=-STEP*SGRD3/UNITGR
      IF(ARG(3).GT.1.2)ARG(3)=1.2
      IF(ARG(3).EQ.1.2)WRITE(6,787)
  787 FORMAT(' ','K3 GREATER THAN 1.2 - LIMITING K3 TO 1.2')
      ARG(1)=SARG1+DLTAK1
      CALL FUNCT(ARG,VAL,GRAD,PIN,PID,PRTLK1,PRTLK2,PRTLK3,PRLDK1,PRLDK2
     C,PRLDK3)
      IF((VAL-SVAL).LT.0.0)GO TO 795
      STEP=STEP/2.0
      IF(STEP.LE.1D-66) GO TO 791
      GO TO 786
  789 WRITE(6,790)
  790 FORMAT(' ','POSSIBLE MINIMUM - GRADIENTS LESS THAN .000001')
      GO TO 793
  791 WRITE(6,792)
  792 FORMAT(' ','STEP SIZE LESS THAN 1D-66 - STOPPING PROGRAM')
      GO TO 793
  793 WRITE(6,794)SVAL,SARG1,SARG2,SARG3,SGRD1,SGRD2,SGRD3,TEST1,TEST2
  794 FORMAT(' ',T4,D11.5,T19,D11.5,T34,D11.5,T49,D11.5,T64,D11.5,T79,D1
     C1.5,T94,D11.5,T109,D8.3,T121,D8.3)
C
C     STATEMENTS BELOW SET VALUES FOR CHARACTERISTIC GUST LENGTH (CL).
C     K1, K2, AND K3 ARE  SET AS ARG(1), ARG(2), AND ARG(3) RESPECTIVELY
C     IMMEDIATELY BELOW THE SPECIFIC CL.
C
  800 CONTINUE
      ILOOP=ILOOP+1
      CL=500.0
      ARG(1)=.5
      IF(ILOOP.EQ.0)GO TO 8
  801 CL=1000.0
      ARG(1)=.5
      IF(ILOOP.EQ.1)GO TO 8
  802 CL=2000.0
      ARG(1)=.5
```

```
      IF(ILOOP.EQ.2)GO TO 8
  803 CL=3000.0
      ARG(1)=.5
      IF(ILOOP.EQ.3)GO TO 8
  804 CL=4000.0
      ARG(1)=.5
      IF(ILOOP.EQ.4)GO TO 8
  805 CL=5000.0
      ARG(1)=.5
      IF(ILOOP.EQ.5)GO TO 8
  806 CL=6000.0
      ARG(2)=325.0
      ARG(1)=1.6
      ARG(3)=-2.60
      IF(ILOOP.EQ.6)GO TO 8
      STOP
      END


      SUBROUTINE FUNCT(ARG,VAL,GRAD,PIN,PID,PRTLK1,PRTLK2,PRTLK3,PRLDK1,
     1PRLDK2,PRLDK3)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ARG(3), GRAD(3)
      DIMENSION PIN(6,6,6), PID(6,6,6)
      DIMENSION PRTLK1(6,6,6), PRTLK2(6,6,6), PRTLK3(6,6,6)
      DIMENSION PRLDK1(6,6,6), PRLDK2(6,6,6), PRLDK3(6,6,6)
      VALN=PIN(1,1,1)
      VALD=PID(1,1,1)
      GRADK1=PRTLK1(1,1,1)
      GRADK2=PRTLK2(1,1,1)
      GRADK3=PRTLK3(1,1,1)
      DGRDK1=PRLDK1(1,1,1)
      DGRDK2=PRLDK2(1,1,1)
      DGRDK3=PRLDK3(1,1,1)
      IF((ARG(1).EQ.0.0).AND.(ARG(2).EQ.0.0).AND.(ARG(3).EQ.0.0))GO TO
     1240
      DO 100 I=1,6
      DO 110 J=1,6
      DO 120 K=1,6
      L=I-1
      M=J-1
      N=K-1
      IF((L.EQ.0).AND.(M.EQ.0).AND.(N.EQ.0))GO TO 120
      IF((L.EQ.0).AND.(M.EQ.0).AND.(N.NE.0))GO TO 10
      IF((L.EQ.0).AND.(M.NE.0).AND.(N.EQ.0))GO TO 20
      IF((L.EQ.0).AND.(M.NE.0).AND.(N.NE.0))GO TO 30
      IF((L.NE.0).AND.(M.EQ.0).AND.(N.EQ.0))GO TO 40
```

```
      IF((L.NE.0).AND.(M.EQ.0).AND.(N.NE.0))GO TO 50
      IF((L.NE.0).AND.(M.NE.0).AND.(N.EQ.0))GO TO 60
      MULTK=(ARG(1)**L)*(ARG(2)**M)*(ARG(3)**N)
      GO TO 150
   10 MULTK=ARG(3)**N
      GO TO 150
   20 MULTK=ARG(2)**M
      GO TO 150
   30 MULTK=(ARG(2)**M)*(ARG(3)**N)
      GO TO 150
   40 MULTK=ARG(1)**L
      GO TO 150
   50 MULTK=(ARG(1)**L)*(ARG(3)**N)
      GO TO 150
   60 MULTK=(ARG(1)**L)*(ARG(2)**M)
  150 IF(PIN(I,J,K).EQ.0.0)GO TO 160
      VALN=VALN+PIN(I,J,K)*MULTK
  160 IF(PID(I,J,K).EQ.0.0)GO TO 180
      VALD=VALD+PID(I,J,K)*MULTK
  180 IF(PRTLK1(I,J,K).EQ.0.0)GO TO 190
      GRADK1=GRADK1+PRTLK1(I,J,K)*MULTK
  190 IF(PRTLK2(I,J,K).EQ.0.0)GO TO 200
      GRADK2=GRADK2+PRTLK2(I,J,K)*MULTK
  200 IF(PRTLK3(I,J,K).EQ.0.0)GO TO 210
      GRADK3=GRADK3+PRTLK3(I,J,K)*MULTK
  210 IF(PRLDK1(I,J,K).EQ.0.0)GO TO 220
      DGRDK1=DGRDK1+PRLDK1(I,J,K)*MULTK
  220 IF(PPLDK2(I,J,K).EQ.0.0)GO TO 230
      DGRDK2=DGRDK2+PRLDK2(I,J,K)*MULTK
  230 IF(PRLDK3(I,J,K).EQ.0.0)GO TO 120
      DGRDK3=DGRDK3+PRLDK3(I,J,K)*MULTK
  120 CONTINUE
  110 CONTINUE
  100 CONTINUE
  240 IF(VALD.EQ.0.0)WRITE(6,1002)VALN,VALD,GRADK1,GRADK2,GRADK3,DGRDK1,
     1DGRDK2,DGRDK3
 1002 FORMAT(' ','VALD EQUAL ZERO PRINTING OUT VALUES CONTINUING ON ',8E
     116.4)
      IF(VALD.EQ.0.0)GO TO 500
      VAL=VALN/VALD
      VALD2=VALD*VALD
      GRAD(1)=((VALD*GRADK1)-(VALN*DGRDK1))/VALD2
      GRAD(2)=((VALD*GRADK2)-(VALN*DGRDK2))/VALD2
      GRAD(3)=((VALD*GRADK3)-(VALN*DGRDK3))/VALD2
  500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ZERO4(DUM1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DUM1(4,4,4)
      DO 100 I=1,4
      DO 110 J=1,4
      DO 120 K=1,4
      DUM1(I,J,K)=0.0
  120 CONTINUE
  110 CONTINUE
  100 CONTINUE
      RETURN
      END


      SUBROUTINE ZERO3(DUM1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DUM1(3,3,3)
      DO 100 I=1,3
      DO 110 J=1,3
      DO 120 K=1,3
      DUM1(I,J,K)=0.0
  120 CONTINUE
  110 CONTINUE
  100 CONTINUE
      RETURN
      END


      SUBROUTINE MULT1(DUM1,DUM2,DUM3)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DUM1(3,3,3), DUM2(2,2,2), DUM3(2,2,2)
      J=1
      K=1
      L=1
      DO 100 I=1,4
      DUM1(J,K,L)=DUM2(J,K,L)*DUM3(1,1,1)+DUM1(J,K,L)
      DUM1(J+1,K,L)=DUM2(J,K,L)*DUM3(2,1,1)+DUM1(J+1,K,L)
      DUM1(J,K+1,L)=DUM2(J,K,L)*DUM3(1,2,1)+DUM1(J,K+1,L)
      DUM1(J,K,L+1)=DUM2(J,K,L)*DUM3(1,1,2)+DUM1(J,K,L+1)
      J=1
      K=1
      L=1
      IF(I.EQ.1)J=2
      IF(I.EQ.2)K=2
      IF(I.EQ.3)L=2
  100 CONTINUE
```

```
      RETURN
      END


      SUBROUTINE MULT2(DUM1,DUM2,DUM3)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DUM1(3,3,3), DUM2(2,2,2), DUM3(2,2,2)
      J=1
      K=1
      L=1
      DO 100 I=1,4
      DUM1(J,K,L)=2.0*DUM2(J,K,L)*DUM3(1,1,1)+DUM1(J,K,L)
      DUM1(J+1,K,L)=2.0*DUM2(J,K,L)*DUM3(2,1,1)+DUM1(J+1,K,L)
      DUM1(J,K+1,L)=2.0*DUM2(J,K,L)*DUM3(1,2,1)+DUM1(J,K+1,L)
      DUM1(J,K,L+1)=2.0*DUM2(J,K,L)*DUM3(1,1,2)+DUM1(J,K,L+1)
      J=1
      K=1
      L=1
      IF(I.EQ.1)J=2
      IF(I.EQ.2)K=2
      IF(I.EQ.3)L=2
  100 CONTINUE
      RETURN
      END


      SUBROUTINE MULT3(DUM1,DUM2,DUM3)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DUM1(4,4,4), DUM2(2,2,2), DUM3(3,3,3)
      J=1
      K=1
      L=1
      DO 100 I=1,4
      DUM1(J,K,L)=DUM2(J,K,L)*DUM3(1,1,1)+DUM1(J,K,L)
      DUM1(J,K,L+1)=DUM2(J,K,L)*DUM3(1,1,2)+DUM1(J,K,L+1)
      DUM1(J,K+1,L)=DUM2(J,K,L)*DUM3(1,2,1)+DUM1(J,K+1,L)
      DUM1(J+1,K,L)=DUM2(J,K,L)*DUM3(2,1,1)+DUM1(J+1,K,L)
      DUM1(J,K+1,L+1)=DUM2(J,K,L)*DUM3(1,2,2)+DUM1(J,K+1,L+1)
      DUM1(J+1,K,L+1)=DUM2(J,K,L)*DUM3(2,1,2)+DUM1(J+1,K,L+1)
      DUM1(J+1,K+1,L)=DUM2(J,K,L)*DUM3(2,2,1)+DUM1(J+1,K+1,L)
      DUM1(J+2,K,L)=DUM2(J,K,L)*DUM3(3,1,1)+DUM1(J+2,K,L)
      DUM1(J,K+2,L)=DUM2(J,K,L)*DUM3(1,3,1)+DUM1(J,K+2,L)
      DUM1(J,K,L+2)=DUM2(J,K,L)*DUM3(1,1,3)+DUM1(J,K,L+2)
      J=1
      K=1
      L=1
```

```
      IF(I.EQ.1)J=2
      IF(I.EQ.2)K=2
      IF(I.EQ.3)L=2
  100 CONTINUE
      RETURN
      END


      SUBROUTINE SUBT1(DUM1,DUM2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DUM1(4,4,4), DUM2(4,4,4)
      DO 100 I=1,4
      DO 110 J=1,4
      DO 120 K=1,4
      DUM1(I,J,K)=DUM1(I,J,K)-DUM2(I,J,K)
  120 CONTINUE
  110 CONTINUE
  100 CONTINUE
      RETURN
      END


      SUBROUTINE SUBT2(DUM1,DUM2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DUM1(3,3,3), DUM2(3,3,3)
      DO 100 I=1,3
      DO 110 J=1,3
      DO 120 K=1,3
      DUM1(I,J,K)=DUM1(I,J,K)-DUM2(I,J,K)
  120 CONTINUE
  110 CONTINUE
  100 CONTINUE
      RETURN
      END


      SUBROUTINE STUFFP(PIX,DHOLD1,DHOLD2,A)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PIX(6,6,6), DHOLD1(3,3,3), DHOLD2(4,4,4)
      DO 600 L=1,10
      I=1
      J=1
      K=1
      IF(L.EQ.1)K=3
      IF(L.EQ.2)J=3
```

```
      IF(L.EQ.3)I=3
      IF(L.EQ.4)I=2
      IF(L.EQ.4)J=2
      IF(L.EQ.5)I=2
      IF(L.EQ.5)K=2
      IF(L.EQ.6)J=2
      IF(L.EQ.6)K=2
      IF(L.EQ.7)I=2
      IF(L.EQ.8)J=2
      IF(L.EQ.9)K=2
      X=DHOLD1(I,J,K)*A
      PIX(I,J,K)=X*DHOLD2(1,1,1)+PIX(I,J,K)
      PIX(I,J,K+1)=X*DHOLD2(1,1,2)+PIX(I,J,K+1)
      PIX(I,J,K+2)=X*DHOLD2(1,1,3)+PIX(I,J,K+2)
      PIX(I,J+1,K)=X*DHOLD2(1,2,1)+PIX(I,J+1,K)
      PIX(I,J,K+3)=X*DHOLD2(1,1,4)+PIX(I,J,K+3)
      PIX(I,J+1,K+1)=X*DHOLD2(1,2,2)+PIX(I,J+1,K+1)
      PIX(I,J+1,K+2)=X*DHOLD2(1,2,3)+PIX(I,J+1,K+2)
      PIX(I,J+2,K)=X*DHOLD2(1,3,1)+PIX(I,J+2,K)
      PIX(I,J+2,K+1)=X*DHOLD2(1,3,2)+PIX(I,J+2,K+1)
      PIX(I,J+3,K)=X*DHOLD2(1,4,1)+PIX(I,J+3,K)
      PIX(I+1,J,K)=X*DHOLD2(2,1,1)+PIX(I+1,J,K)
      PIX(I+1,J,K+1)=X*DHOLD2(2,1,2)+PIX(I+1,J,K+1)
      PIX(I+1,J,K+2)=X*DHOLD2(2,1,3)+PIX(I+1,J,K+2)
      PIX(I+1,J+1,K)=X*DHOLD2(2,2,1)+PIX(I+1,J+1,K)
      PIX(I+1,J+1,K+1)=X*DHOLD2(2,2,2)+PIX(I+1,J+1,K+1)
      PIX(I+1,J+2,K)=X*DHOLD2(2,3,1)+PIX(I+1,J+2,K)
      PIX(I+2,J,K)=X*DHOLD2(3,1,1)+PIX(I+2,J,K)
      PIX(I+2,J,K+1)=X*DHOLD2(3,1,2)+PIX(I+2,J,K+1)
      PIX(I+2,J+1,K)=X*DHOLD2(3,2,1)+PIX(I+2,J+1,K)
      PIX(I+3,J,K)=X*DHOLD2(4,1,1)+PIX(I+3,J,K)
  600 CONTINUE
      RETURN
      END
```

LIST OF REFERENCES

1. Blakelock, J. H., Automatic Control of Aircraft and Missiles, John Wiley and Sons, Inc., 1965.

2. Stapleford, R. L., and others, A Practical Optimization Design Procedure for Stability Augmentation Systems, AFFDL-TR-70-11, October 1970.

3. Hess, R. A., An Application of Optimal Stochiastic Control Theory to the Problem of Aircraft Gust Alleviation, Ph.D. dissertation, Aug. 1970, University of Cincinnati, Cincinnati, Ohio.

4. Etkin, R., Dynamics of Flight, John Wiley and Sons, Inc., 1963.

5. Newton, G. C., Gould, L. A. and Kaiser, J. F., Analytical Design of Linear Feedback Systems, John Wiley and Sons, Inc., 1957.

6. Hess, R. A., Some Results of Suboptimal Gust Alleviation, AIAA Journal of Aircraft, to be published.

7. McClean, R., The Optimization of an Autopilot for an Airplane Subjected to Random Atmospheric Turbulence, UTIA Technical Note No. 45, Nov. 1960.

8. Hart, J. E., Adkins, L. A., and Lacau, L. L., Stochiatic Disturbance Data for Flight Control System Analysis, ASD-TDR-62-347, Sept. 1962.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Dr. Ronald A. Hess, Code 57He Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	LCDR Russell L. Clement SMC Box 1704 Naval Postgraduate School Monterey, California 93940	2

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE

A Parameter Optimization Approach to Aircraft Gust Alleviation

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Master's Thesis; March 1972

5. AUTHOR(S) *(First name, middle initial, last name)*

Russell Lee Clement

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1972	64	8

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

This research indicates that a knowledge of angle of attack (α) and/or gust perturbation angle of attack (α_g) is of prime importance for a longitudinal autopilot designed to alleviate the effect of vertical gusts. However, accurate measurement of α and α_g is extremely difficult. This research indicates that the readily measurable normal acceleration factor (n) and pitch rate ($\hat{q}$) can be used to provide α and α_g information indirectly.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

UNCLASSIFIED

Security Classification

A-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Aircraft Gust Alleviation						
Longitudinal Autopilot						
Parameter Optimization of Autopilot						
Use of Random Gust Fields						
Autopilot Design						

Thesis 134466
C5093 Clement
c.1 A parameter optimization approach to aircraft gust alleviation.